

LONG SHORT-TERM MEMORY NEURAL NETWORKS FOR CLOGGING DETECTION IN THE SUBMERGED ENTRY NOZZLE

Ana P. M. Diniz¹, Patrick M. Ciarelli¹, Evandro O. T. Salles¹, and Klaus F. Coco¹

¹ Universidade Federal do Espírito Santo, Vitória, Espírito Santo, Brazil

Received: 2 April 2022;
Accepted: 8 May 2022;
Available online: 13 May 2022.

Original scientific paper

Abstract: *The clogging in the Submerged Entry Nozzle (SEN), responsible for controlling the steel flow in continuous casting, is one of the main problems faced by steelmaking process, since it can increase the frequency of interruptions in the operation for the maintenance and/or exchange of its equipment. Although it is a problem inherent to the process, not identifying the clogging can result in losses associated with the process yield, as well as compromising the product quality. In order to detect the occurrences of clogging in a real steel industry from historical data of process variables, in this paper, different models of Long Short-Term Memory (LSTM) neural networks were tested and discussed. The overall performance of the classifiers developed here showed very promising results in real data with class imbalance.*

Key words: *Continuous Casting, Submerged Entry Nozzle, Clogging, LSTM, Deep Learning.*

1. Introduction

One of the problems faced by the industry concerning the continuous casting process is the accumulation of steel impurities that forms on Submerged Entry Nozzle (SEN) of the tundish, causing their obstruction, known as clogging (Ikaheimonen, 2002).

As evidenced by Ikaheimonen (2002), clogging can be formed by several factors, including metallurgical, hydrodynamic and thermodynamic factors, as well as nozzle material, unpredictable disturbances and operational failures.

According to Rackers (1995), the consequences of clogging include reduced productivity, increased production costs and decreased product quality. Clogging events increase the frequency of interruptions in the operation for the exchange and/or maintenance of nozzles and tundishes, which can reduce the useful lifetime by

* Corresponding author.

E-mail addresses: ana.diniz@edu.ufes.br (A. P. M. Diniz), patrick.ciarelli@ufes.br (P. M. Ciarelli), evandro@ufes.br (E. O. T. Salles), klaus.coco@ufes.br (K. F. Coco)

up to half (Schmidt, Russo & Bederka, 1991). In addition, small solid inclusions can break off and enter the steel flow, causing unacceptable defects in the product (Bessho et al., 1991, Wang et al., 2021, Wu et al., 2021).

Scientific studies show that the clogging phenomenon lasts around 250 seconds, being perceptible only after the first 80 seconds, which leaves a margin of 170 seconds between possible detection and total obstruction (Barati et al., 2018). Thus, the detection model must act quickly to allow corrective action in less than 2 minutes.

Although the detection of the beginning of clogging is of fundamental importance so that control actions can be applied and the system operation is prolonged, in practice, the steel industry still does not have effective tools for the detection (Rout et al., 2013, Pellegrini et al., 2019, Wang et al., 2021). Due to hostile working conditions, variations in production, process and sensors failures, the data are generally noisy, with outliers and missing values (Wang, Mao & Huang, 2018).

Even with the different adversities, researches have been developed and, usually, they correlate the clogging occurrences with the gate opening and the casting speed. There is not a large number of studies found in the bibliography, mainly due to the difficulty in obtaining the data due to their confidentiality. In addition, these researches apply techniques associated with the physical parameterization of the plant, which restricts their applicability (Ikäheimonen et al., 2002, Vannucci & Colla, 2011, Rout et al., 2013, Pellegrini et al., 2019).

Regarding the models developed in the literature, in addition to being heavily influenced by physical equations, the small number of clogging occurrences against normal operating data end up compromising the classifiers' ability during the learning phase. As a result, the predictive accuracy desired by the industry can hardly be guaranteed (Vannucci & Colla, 2011, Rout et al., 2013, Pellegrini et al., 2019).

Several works in this area were evaluated and the best success rates for clogging prediction were found by Vannucci & Colla (2011) and Pellegrini et al. (2019). The first authors associated neural networks with fuzzy logic to classify clogging events, achieving 76.9% of recall and 80.2% of accuracy. In Pellegrini et al. (2019) were applied an online predictive estimation model for the probability of clogging using about 50 process variables. Although the authors suggest that the model presented a good classification performance in series identified as possibly subject to clogging, with an overall area under the curve (AUC) equal to 0.8, when tested in series with lower probability of clogging incidence, the model presented an accuracy of 75% and a precision of 62%.

In Ikäheimonen et al. (2002), neural networks were applied to data from a real plant in a problem similar to the one addressed in this study, however, satisfactory results were not obtained in the interest of the industry. Another aspect is related to the amount of noise inherent to the input signals, so that a multilayer perceptron neural network did not behave so well in the initial clogging prediction task. The idea is to apply as few pre-processing techniques as possible, in order to be able to use real data in an online application.

As discussed by Goodfellow, Bengio & Courville (2016), the use of deep learning is motivated by the difficulty of traditional algorithms in generalizing problems involving, above all, high-dimensional and highly complex data. Deep learning, then, provides a very powerful framework for supervised learning (Wang et al., 2021, Wu et al., 2021).

In order to identify the clogging in the SEN, this article evaluates the general performance of classifiers using Long Short-Term Memory (LSTM) neural networks. This type of algorithm has been used as an important tool in several researches for

extracting temporal resources from sequential data (Yildirim et al., 2019, Essien & Giannetti, 2020, Wang et al., 2021, Wu et al., 2021). These factors motivated us to apply deep neural networks, such as LSTM, which is capable of extracting information relevant to clogging detection even with a high-noise signal.

Therefore, this study is motivated by the contribution in the application of techniques exclusively based on data for the detection of the initial occurrence of obstruction, since the recent researches are based on idealized systems and lack sufficient precision in complex tasks or dynamic environments.

In general, the performance of the classifiers developed here showed very promising results in real data applications, obtaining precision and recall levels above 85%. The correct classification of clogging occurrences can contribute to reducing process interruptions and costs associated with production, as well as improving the quality of the final product (Vannucci & Colla, 2011, Pellegrini et al., 2019, Wang et al., 2021).

This article is divided as follows: Section 2 discusses the causes and effects of the occurrence of clogging in the continuous casting process. Section 3 presents LSTM neural network. In Section 4, the dataset is presented together with the step-by-step of the proposed methodology, as well as the performance metrics used in the classification task. In Section 5, the results are presented, comparing the performance of the classifiers, followed by Section 6 that provides the final considerations and conclusions.

2. Clogging in the Continuous Casting Process

The continuous casting process is based on the vertical casting of liquid steel from a ladle positioned on a tundish. In Figure 1, a typical schematic of the steel flow from the tundish to the mold is presented. The steel flows through the Tundish Nozzle, being regulated by the Slide Gate and introduced into the Copper Mold cavity through the SEN and Nozzle Port. Thus, its flow process begins to solidify (Rackers, 1995, Mourão et al., 2011).

The SEN has a fundamental role in the stability of the process and quality of the final product, being fundamental in the production of special steels (Rackers, 1995). However, throughout the process, an accumulation of impurities from the steel builds up on the nozzle wall, developing the clog. As the obstruction increases, the Slide Gate must be opened in order to maintain the desired flow. However, when its opening reaches 100%, production must stop and the SEN or even the Tundish Set (composed of the Tundish, Slide Gate and SEN) must be replaced in advance (Thomas & Bai, 2001).

From the prototype model used to simulate the casting speed variation by solid deposition over time, Barati et al. (2018) established three stages for the formation of clogging. Throughout the process, deposition of particles occurs in the SEN. When the clogging event starts, during the first 80 seconds, some regions of the middle section of the SEN are covered by a smooth layer of clogging (coverage stage). Then, there is the bulging phase in which the deposition of particles occurs more quickly, emerging visible particles. This phase occurs up to about 150 seconds and is followed by the branching step where there is the development of a branched structure that grows continuously until the SEN cross-section is completely blocked, around 250 seconds. In general, as this phenomenon occurs only after some heats, it is not necessary the SEN to be fixed or cleaned at such a high frequency.

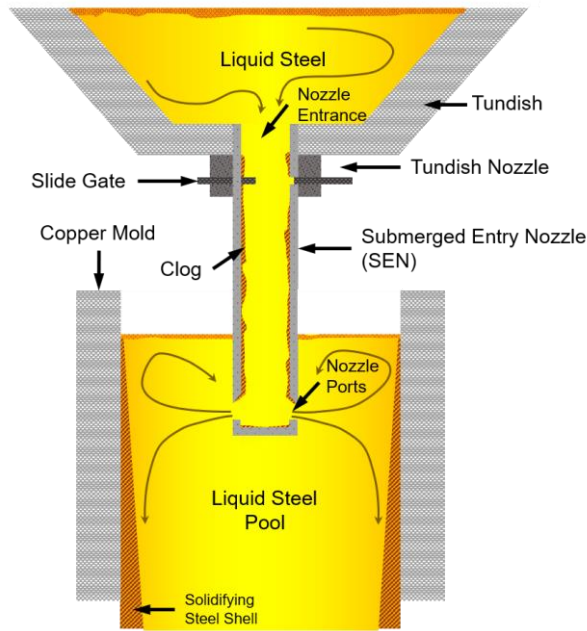


Figure 1. Schematic summary of the flow of steel from the tundish to the mold with flow control performed by the Slide Gate

In their research, Mourão et al. (2011) found that clogging can be formed not only by solidified steel and the transport of oxides present in it, but also by the aspiration of air in the SEN and the chemical reactions. However, they emphasize that the exact causes of clogging, specifically, can be difficult to identify.

3. Long Short-Term Memory (LSTM)

The LSTM has a set of recurrently connected memory blocks. Each block contains one or more interconnected cells and three multiplicative units, also called forget gate $f(t)$, input gate $i(t)$ and output gate $o(t)$ (Haykin, 2011, Goodfellow, Bengio & Courville, 2016, Buduma & Locascio, 2017).

The basic architecture of an LSTM cell is shown in Figure 2, where $x(t)$ corresponds to the input signal, $C(t)$ and $C(t-1)$ are, respectively, the current state of the memory cell and its previous instant and $h(t)$ and $h(t-1)$ represent its current and previous hidden state, respectively.

The signals are sent to the three gates that control the information. The function of forget gate $f(t)$ is to control which parts of the long-term states should be forgotten. The input gate $i(t)$, in turn, has the function of controlling which parts should be added to the long-term states. The output gate $o(t)$ is responsible for controlling the output information $h(t)$ in the current state of time. The gates outputs are calculated using:

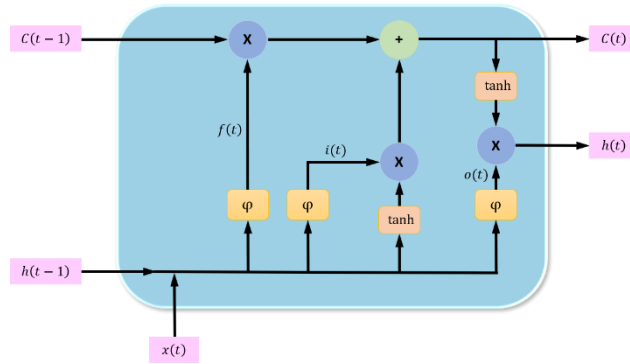


Figure 2. The structure of an LSTM cell

$$f(t) = \varphi(w_{fx} \cdot x(t) + w_{fh} \cdot h(t - 1) + b_f) \quad (1)$$

$$i(t) = \varphi(w_{ix} \cdot x(t) + w_{ih} \cdot h(t - 1) + b_i) \quad (2)$$

$$o(t) = \varphi(w_{ox} \cdot x(t) + w_{oh} \cdot h(t - 1) + b_o) \quad (3)$$

where $\varphi(\cdot)$ is a nonlinear activation function that, in general, uses the sigmoid function. Thus, the updates of the state of the memory cell $C(t)$ and of the hidden state $h(t)$ are generated, respectively, by

$$C(t) = f(t) \odot C(t - 1) + i(t) \odot \tanh(w_{cx} \cdot x(t) + w_{ch} \cdot h(t - 1) + b_c) \quad (4)$$

$$h(t) = o(t) \odot \tanh(c(t)) \quad (5)$$

where $\tanh(\cdot)$ represents the hyperbolic tangent activation function and \odot denotes the point multiplication operation between two vectors. The terms w_{fx} , w_{ix} , w_{ox} and w_{cx} correspond to the input weights of each gate, while w_{fh} , w_{ih} , w_{oh} and w_{ch} refer to their respective recurrent weights and the terms b_f , b_i , b_o and b_c represent the bias.

LSTM avoids the disappearance of the gradient through the switch of its gates, which develop a kind of temporal memory. During the training phase, samples from each batch are passed into cells iteratively through states. The hidden state represents short-term memory, while the cell state is long-term memory. It is in this unit that information is propagated through the network, interacting with the cell through the ability to remove or add information through gates. As a result, they are able to identify which temporal information should be transmitted or discarded by the network. After processing each batch, the internal states of each cell are reset (Goodfellow, Bengio & Courville, 2016, Buduma & Locascio, 2017).

The great complexity of networks based on deep learning can lead to a problem known as overfitting, thus, it is common to use a regularization technique called dropout. In it, at each training iteration, there is a random removal of a pre-established percentage of neurons from a given layer, adding them again in the next iteration. Considering that a given neuron will not depend on the specific presence of the others, dropout enables the learning of the network to deal with more robust attributes (Goodfellow, Bengio & Courville, 2016, Buduma & Locascio, 2017).

4. Dataset and Methodology

4.1. Dataset

In this paper, historical data from 6 months of measurements made in a continuous casting steelmaking process were used. The variables were collected from two tundishes operating on a mold at a rate of one sample per second. It is important to clarify that in this article does not present data, nor specific characteristics of the industrial process, as well as the company name due to the data confidentiality protocol.

Ideally, the process specialists classify as clogging the corresponding event to the gate opening without increase the mold level, with or without a variation in casting speed. In most cases, the gate opening occurs after increasing oscillations, which may or may not reflect oscillations in level. The nature of the data, however, does not allow the use of simple rules to classify clogging occurrences. For example, there are clogging events where the casting speed is increasing while the gate opening has a much higher rate than expected.

Figure 3 illustrates an example of the clogging event through the process. Starting from sample 10.601, there is a gradual increase in the gate opening, at constant speed casting, without a significant increase in the mold level. However, due to the reduction of the SEN section caused by the obstruction, a small increase in level is observed just before the casting speed is reduced. The reduction of casting speed provides a gate closure and, when the casting speed reaches a level below 0.6 m/s, the level starts to rise. It is also observed that during this period there is no exchange of tundishes, since the indicative variable of tundish in operation remains constant.

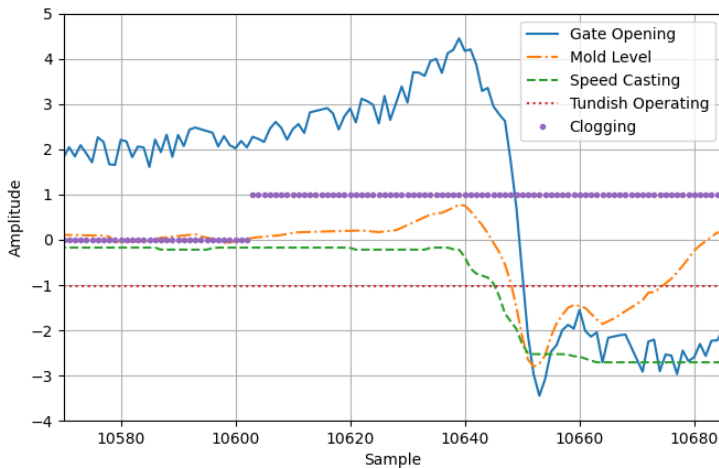


Figure 3. Example of clogging occurrence observed from selected process variables

Thus, based on experiments performed in the literature (Ikaheimonen, 2002, Vannucci & Colla, 2011, Pellegrini et al., 2019) four process variables were selected: the percentage of the slide gate's total opening (gate opening), mold level, casting speed and the tundish that is operating (tundish operating). Researchers also suggest

the use of variables related to temperature, pressure and argon flow; however, we did not observe in our dataset a significant correlation between these variables and the studied phenomenon. This may have happened because of the high number of outliers in the dataset. Furthermore, it was observed that these variables were not effectively measured in the period considered, possibly due to sensor failures.

After selecting the variables, pre-processing was applied for the treatment of outliers and missing data, ensuring the preservation of the relationships between the attributes.

The outliers' occurrences were seen as measurement errors because they are specific cases and, therefore, were treated in order to make them consistent. For this purpose, the maximum and minimum theoretical values assumed by each of the variables during an operation without anomalies were specified. Thus, samples with values outside the theoretical range were considered outliers: if a sample had a value below the theoretical minimum, then it was adjusted to the theoretical minimum value assumed by the variable. Analogously, if the assumed value was greater than the theoretical maximum, then it was set to its theoretical maximum value.

In relation to missing data, no substantial occurrences were identified in the period and variables analyzed. Still, the few absence periods longer than a sample were treated by a moving median filter.

Furthermore, a certain unbalance between the clogging and non-clogging classes was verified, which was expected for this type of problem, since clogging is a failed event. It was observed that a little more than 7% of the samples indicated the occurrence of the event. Given the number of available samples, the data were selected in order to promote the balance between the classes of the training data only, without losing great importance information for the modeling.

In Barati et al. (2018) were observed that clogging is only truly perceived in the SEN section between approximately 100 and 150 seconds. Therefore, as it is an autoregressive system, the composition of the input matrix was made through a sliding window covering 120 samples and with a step of 1 sample per iteration. In this way, the past behavior of each variable will be used to classify whether in the present there is clogging or not.

Figure 4 illustrates a time-based sliding window process. Following this analysis, given an output, $y(t)$, corresponding to one of the binary classes at time t , it will be related to the input matrix, $X_{(120 \times 4)} = \{x_1, x_2, x_3, x_4\}^T$, containing the 4 selected variables with samples delayed 120-time steps, i.e., from $(t-119)$ to t . Thus, each matrix formed will be labeled with one of the classes, that is, the event (clogging or non-clogging) that is happening at instant t . The sliding window moves along the series so that, at each step, a new input matrix and its respective class are generated.

The dataset after pre-processing is composed of 422,026 matrices of dimensions 4×120 , i.e., 120 samples of each of the four input variables (gate opening, speed casting, mold level and tundish operating). The dataset was divided into training, validation and testing, with 70% of the data applied in training ($N_{tr} = 292,170$ matrices), 17% for validation ($N_{val} = 73,044$ matrices) and 13% reserved for the testing stage ($N_{est} = 56,812$ matrices).

Long Short-Term Memory Neural Networks for Clogging Detection in the Submerged Entry Nozzle

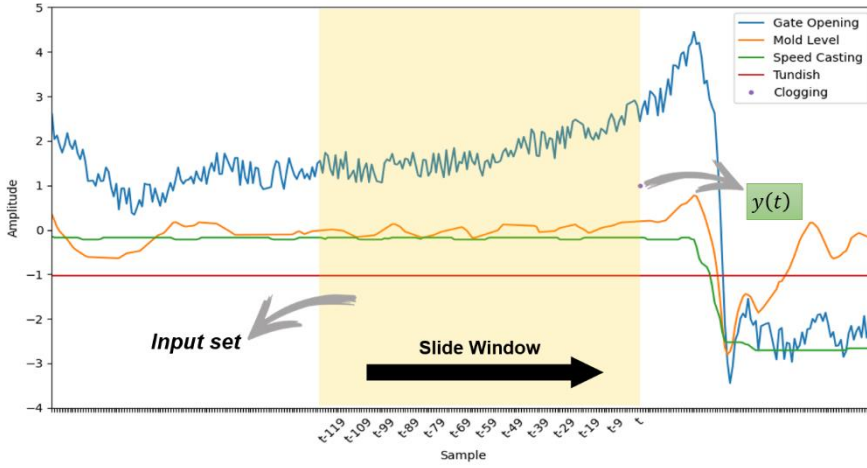


Figure 4. Sliding Window Process

As shown in Table 1, each training matrix has a corresponding class, which is distributed in 50% representing the clogging event and the others representing the non-clogging class. On the other hand, the validation and test datasets remain with the old proportion between classes.

Data standardization (z-score) was performed for each variable of each set according to Eq. (6) so that each variable has zero mean and unitary standard deviation. In the equation, $x_r(t)$ represents the variable to be standardized, r (where $r \in \{1,2,3,4\}$) is the index identifying each variable, μ_r is the mean of $x_r(t)$ and σ_r is the respective standard deviation (both μ_r and σ_r were computed using training data). Standardization was chosen because it better handles possible outliers present in the series (Skiena, 2017).

$$x_{z-score}(t) = \frac{x_r(t) - \mu_r}{\sigma_r} \quad (6)$$

Table 1. Dataset split and proportion of each class.

Dataset	Number of Matrices	Split Proportion (%)	Clogging Class (%)	Normal Class (%)
Training	292,170	70	50	50
Validation	73,044	17	93	7
Tests	56,812	13	93	7

4.2. Methodology

In structural terms, different parameter configurations, defined by the trial-and-error method, were applied to the LSTM classifiers. For this purpose, only the number of cells in the two LSTM layers was varied in increments of eight, using all their states. In turn, a Fully Connected (FC) layer with 200 neurons was always maintained at the output of the last LSTM layer.

The four input variables, lagged by 120 samples each, were applied to the network using a Batch Size of 1,200 samples. A limit of 200 epochs for training was chosen,

which can be interrupted by early stopping with patience of 5. The weights were updated using Adam Algorithm with learning rate of 0.001 and Cross-Entropy as loss function (Kingma & Ba, 2014, Goodfellow, Bengio & Courville, 2016). Due to the stochastic nature of the models used here, the k-fold was applied, with $k = 5$, for validation and comparison of the classifiers.

From then on, eight classifier configurations presented the best performances in the training phase from the validation set. Table 2 shows the configurations of each classifier and also the number of trained parameters.

Table 2. The best-found LSTM models and their main parameters associated with their architectures.

LSTM Model	LSTM Layer 1 Cells	LSTM Layer 2 Cells	Number of Parameters
1	256	128	490.586
2	256	64	362.842
3	256	32	311.258
4	256	16	288.538
5	256	8	277.946
6	128	64	130.906
7	128	32	95.706
8	64	32	37.082

Figure 5 represents a schematic diagram of the structure of the networks used in this paper.

The LSTM networks were implemented with two layers, each containing 32 to 256 cells, with hyperbolic tangent function. It is interesting to mention that, although network configurations containing 32 cells in the first LSTM layer were tested, relevant results were not obtained. The dropout was applied to the second LSTM layer with a value of 0.3.

In classification tasks, the number of output layer neurons will be equal to the number of classes to be predicted. In this layer, it is common to use the Softmax activation function. The Softmax function helps to map the output of a neural network to a set of categories, by transforming these responses into probabilities that add up to 1. The Softmax function is calculated as follows, where J is the number of classes:

$$\text{softmax}(y_j) = \frac{\exp(y_j)}{\sum_{i=1}^J \exp(y_i)} \quad (7)$$

where y_i is the i -th neural network output ($i = 1, \dots, J$) and y_j is the class whose categorical probability is calculated in the equation.

In this way, in all models the signal was transferred to a FC layer containing 200 neurons, activation function Rectified Linear Unit (ReLU) and dropout of 0.5, concatenated to a Softmax layer with 2 neurons to generate the probability of classifying the input time series as clogging or non-clogging.

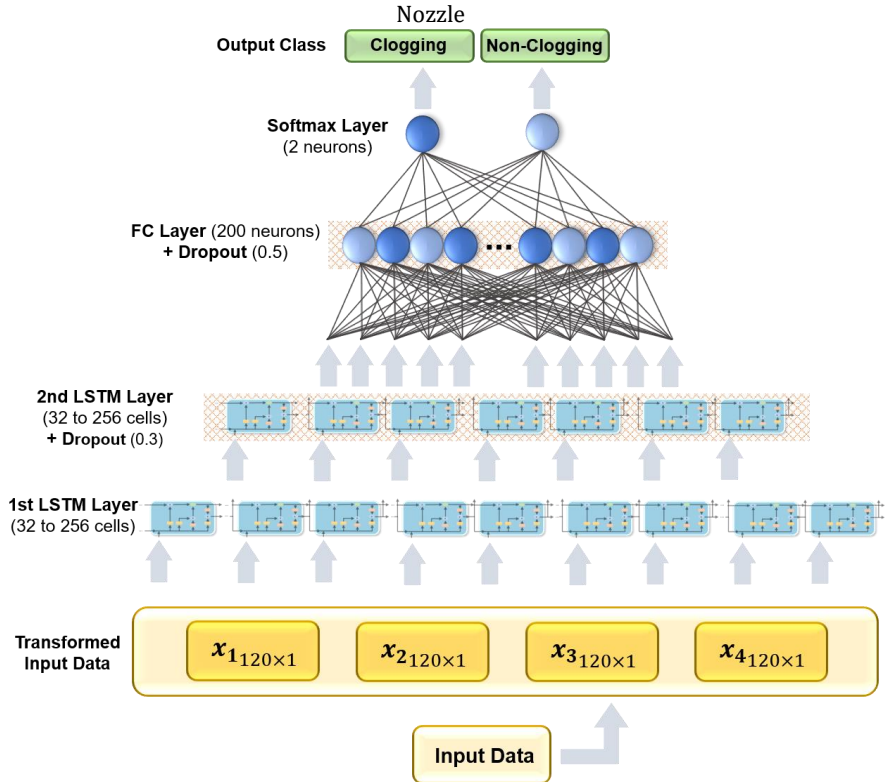


Figure 5. Generic structure of the LSTM classifier used in this article

The simulations were implemented on a computer in an environment Python 3.6, 64 bit operating system, 16 GB of RAM, Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz 2.59 GHz with GPU NVIDIA GeForce Titan XP.

4.3. Performance Criteria in Classification Tasks

The confusion matrix is usually used in the analysis of performance in classification tasks. As seen in Figure 6, it is the result of comparing the correct class of each sample in the test set and the class obtained by the classifier.

		Predicted Class	
		Positive	Negative
Actual Class	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Figure 6. Confusion Matrix

In binary classification tasks, the confusion matrix is composed of positive and negative class observations (Fawcett, 2016). In this work, the occurrence of clogging will be associated with the positive class and its absence with the negative class. Thus, after classification, the values can belong to four possible categories:

- True Positive (TP): samples that belong to the positive class that were correctly classified;
- False Positive (FP): samples that belong to the negative class, but they were incorrectly classified as belonging to the positive class;
- True Negative (TN): samples that belong to the negative class that were correctly classified;
- False Negative (FN): samples that belong to the positive class, but they were incorrectly classified as belonging to the negative class.

From these categories, some performance measures can be calculated, such as accuracy, precision and recall, respectively, according to Eq. (8), Eq. (9) and Eq. (10).

$$accuracy = \frac{TP+TN}{(TP+FN+TN+FP)} \quad (8)$$

$$precision = \frac{TP}{(TP+FP)} \quad (9)$$

$$recall = \frac{TP}{(TP+FN)} \quad (10)$$

Accuracy measures the overall performance of the model, considering both the proportion of correct classifications of positive and negative cases. In contrast, precision measures the rate of positive examples classified correctly among all those predicted as positive. Recall corresponds to the rate of classification of true positives in relation to the total number of positive examples (Fawcett, 2016).

From the combination of accuracy and recall, it is possible to obtain an indicator of the overall quality of the model, called F1-Score. The F1-Score calculation is represented in Eq. (11).

$$F1 = \frac{2 \times precision \times recall}{(precision+recall)} \quad (11)$$

In some cases, it is also interesting to evaluate the Matthews Correlation Coefficient (MCC). The MCC is a more reliable and balanced measure of quality, as it only produces a high score if the classifier correctly predicted most of the positive and negative grades. Its equation is presented in Eq. (12) (Boughorbel, Jarray & El-Anbari, 2017).

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP+FP) \times (TP+FN) \times (TN+FP) \times (TN+FN)}} \quad (12)$$

The MCC varies in the range between -1 and 1, where the extreme values +1 and -1 indicate, a perfect classification and a totally incorrect classification, respectively, while the value 0 indicates a classification equivalent to what would be done randomly (Boughorbel, Jarray & El-Anbari, 2017).

5. Results

The performances of the eight LSTM models are shown in Table 3 in terms of mean and standard deviation of accuracy, precision, recall, F1-Score and MCC.

As we can see, the LSTM 1, LSTM 2 and LSTM 6 models reached levels of accuracy and precision above 80%. However, these were also the models that showed the greatest standard deviations. In particular, the LSTM 1 model stands out for its 85.30% recall, that is, around 5% more than the other two models. The same can be observed for the MCC metric, where the LSTM 1 model also stands out, reaching an average of 0.723 with a standard deviation around 0.145. The expected MCC metric for a Dummy classifier, which is based on the majority class, is close to zero. Therefore, the obtained results are superior to a Dummy classifier.

Table 3. Comparison among the performances of different LSTM models..

LSTM Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	MCC
1	86.10±3.87	85.61±3.42	85.30±5.51	85.45±4.22	0.723±0.145
2	82.40±5.11	83.09±5.52	79.71±5.15	81.36±5.33	0.631±0.161
3	75.81±0.31	79.55±0.65	69.49±1.08	74.18±0.81	0.520±0.006
4	74.31±0.68	77.69±2.67	68.52±3.22	72.82±2.92	0.491±0.016
5	75.20±0.41	81.84±1.02	64.79±0.56	72.32±0.72	0.515±0.010
6	82.02±4.33	82.54±7.41	80.89±7.47	81.32±7.44	0.612±0.085
7	75.83±0.19	79.79±0.80	69.22±1.47	74.13±1.04	0.521±0.013
8	75.76±0.38	79.15±0.81	69.95±0.87	74.27±0.84	0.518±0.080

Due to the nature of the process, noise and, mainly, control actions by operators may be present in measurements, making it difficult to use simple rules assertively. Furthermore, due to the unbalance between classes, although this strategy can result in high rates of global accuracy, the Dummy classifier would hinder the identification of examples belonging to rare classes that, in the problem in question, represent the interest class.

Although precision and recall had values close to each other in the three main models (LSTM 1, LSTM 2 and LSTM 6), it is interesting to note that the average recall did not exceed precision. In accordance with Eq. (9) and Eq. (10) this result indicates a number of FN higher than the number of FP.

In general, it is possible to observe that the reduction in the number of parameters of the second layer compromised the generalizability of the models, mainly in terms of recall, and consequently, F1-Score. For example, the LSTM 1 and LSTM 2 models differ by the number of cells in the second layer and, as shown in Table 3, the LSTM 1 model, which has the highest degree of complexity among the models, showed greater generalization capacity in the classification task.

In practice, models analysis must be based on a trade-off between the values of the performance metrics and the number of parameters involved. Comparing the parameters used by LSTM 1 and LSTM 2 models with those of the LSTM 6 model, which also achieved promising results, it appears that there are about 3.74 more parameters in the first model and 2.77 in the second model. In particular, it is observed that the overall performance of the LSTM 2 and LSTM 6 models does not change considerably, which makes the LSTM 6 model more attractive.

Furthermore, although a small reduction in training time was observed with the decrease in the number of parameters, no significant differences in processing time were observed during the tests of the models.

Since this is a problem involving real process data, methodologies capable of correctly detecting the presence of clogging with the lowest possible error rate are sought. In this context, the LSTM 1 model appears to be the best choice, since it presents the highest performance averages and the smallest deviations from these averages (among the LSTM 1, LSTM 2, and LSTM 6 models). Even taking into account the complexities of the classifiers, the models with the highest number of parameters still seem to be the most attractive, since the training and processing time of these models did not show a significant increase compared to the others.

In addition, the LSTM 1 model exceeded the 75% accuracy and 62% precision of Pellegrini et al. (2019), as well as the 76.9% recall of Vannucci & Colla (2011). Although the significant differences in methodologies and datasets, it can be said that the LSTM 1 model achieved promising levels of performance criteria in the clogging classification task compared to those found in the bibliography.

6. Conclusion

The steel industry does not have effective tools that can correctly detect the occurrences of clogging in the SEN. Clogging can increase the frequency of interruptions in the operation, resulting in an increase in operating costs, decreased productivity, and adversely affect product quality.

In order to treat this problem, in this paper, the general performance of classifiers using LSTM neural networks in detecting the clogging in the SEN of a steel production process were evaluated. Therefore, based on the evidences observed in the literature and analyzes made in this work, four variables were selected to compose the input dataset: gate opening, mold level, casting speed and tundish operating.

The dataset was then pre-processed in order to deal with the presence of outliers and missing values, ensuring the relationships between the attributes. Furthermore, due to the inherent unbalance of the classes, a careful balancing step was necessary for the training, so that the information of great importance for the modeling was not disregarded. With balanced classes, accuracy can be considered in the evaluation of models, taking into account also other metrics: precision, recall, F1-Score and MCC.

Industrial problems require high hit rates and, within the task of classifying clogging occurrences, the model with the highest number of parameters obtained a remarkably superior performance in relation to the other evaluated models, presenting the highest performance averages. In particular, the best model achieved precision and recall averages above 85%. However, even with higher values of mean precision and recall, a higher number of False Negatives was found in relation to the number of False Positives, considering that the recall did not exceed precision. This characteristic demonstrates that the best model found is more probable to classify a sample referring to the occurrence of clogging (positive) as normal operation (negative) than the reverse.

Nevertheless, given the significant differences in methodologies and datasets, it can be said that the best model achieved promising levels of performance criteria in the task of classifying clogging compared to those found in the literature.

Through the proposed methodology, the possibility of applying the models in a real system was verified. However, the model presented is limited in terms of recall and precision. The recall values did not exceed the precision values, which indicates the

need to develop other techniques to reduce false negative rates. For future work, it is desirable to implement other types of classifiers with deep learning architectures, such as ConvLSTM, so that even higher hit rates can be achieved and guaranteeing a reduction in FP rates compared to FN rates. It is also intended to apply the models online in a real plant.

The fruits of this work may favor productivity gains and reduction of production costs, due to the increase in SEN's useful life, for example. It is estimated that the application of a system that allows the identification of the initial occurrence of clogging can contribute to the increase in the useful life of the SEN, which may result in a savings of over US\$ 15 million per year.

Author Contributions: All authors actively participated in all stages of this research.

Funding: This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Data Availability Statement: This article does not present data, nor specific characteristics of the industrial process, as well as the company name due to the data confidentiality protocol.

Acknowledgments: The authors would like to thank the financial support provided by CAPES, as well as the support of the Programa de Pós-Graduação em Engenharia Elétrica (PPGEE). The authors would also like to thank NVIDIA Corporation for the donation of a Titan XP GPU used for this research.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- Barati, H., Wu, M., Kharicha, A. & Ludwig, A. (2018). A transient model for nozzle clogging. *Powder Technology*, 329, 181-198.
- Bessho, N., Yoda, R., Yamasaki, H., Fujii, T., Nozaki, T. & Takatori, S. (1991). Numerical analysis of fluid flow in continuous casting mold by bubble dispersion model. *ISIJ international*, 31(1), 40-45.
- Boughorbel, S., Jarray, F. & El-Anbari, M. (2017) Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric. *PloS one*, 12(6), e0177678.
- Buduma, N. & Locascio, N. (2017). *Fundamentals of deep learning: Designing next generation machine intelligence algorithms*. Sebastopol: O'Reilly Media, Inc.
- Essien, A. & Giannetti, C. (2020). A deep learning model for smart manufacturing using convolutional LSTM neural network autoencoders. *IEEE Transactions on Industrial Informatics*, 16(9), 6069-6078.
- Fawcett, T. (2016). An introduction to ROC analysis, *Pattern recognition letters*, 27(8), 861-874.
- Goodfellow, I., Bengio, Y. & Courville, A. (2016). *Deep Learning*. Cambridge: MIT Press.

Haykin, S. O. (2011). *Neural Networks and Learning Machines*. New Jersey: Pearson Education.

Ikaheimonen, J., Leiviska, K., Ruuska, J. & Matkala, J. (2002). Nozzle clogging prediction in continuous casting of steel. *IFAC Proceedings Volumes*, 35(1), 143-147.

Kingma, D. P. & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Mourão, M. B., Yokoji, A., Malynowskyj, A., Leandro, C. A. S., Takano, C., Quites, E. E. C., Gentile, E. F., Silva, G. F. B. L., Bolota, J. R. & Gonçalves, M. (2011). *Introdução à Siderurgia*. São Paulo: Associação Brasileira de Metalurgia, Materiais e Mineração.

Pellegrini, G., Sandri, M., Villagrossi, E., Challapalli, S., Cestari, L., Polo, A. & Ometto, M. (2019). Successful use case applications of artificial intelligence in the steel industry. In: *Iron & Steel Technology (AIST), Digital Transformations*, 44-53.

Rackers, K. G. (1995). *Mechanism and mitigation of clogging in continuous casting nozzles*. Master's thesis, University of Illinois at Urbana-Champaign.

Rout, B. K., Singh, R. K., Choudhary, S. K. & Das, C. L. (2013). Development and application of nozzle clogging index to improve the castability in continuous slab casting. In: *International Conference on Advances in refractories and Clean Steel Making*, At Ranchi, India.

Schmidt, M., Russo, T. J. & Bederka, D. J. (1991). Steel shrouding and tundish flow control to improve cleanliness and reduce plugging. *Iron and Steel Society, Tundish Metallurgy*, 2, 3-12

Skiena, S. S. (2017). *The data science design manual*. New York: Springer.

Thomas, B. G. & Bai, H. (2011). Tundish nozzle clogging-application of computational models. In *Steelmaking Conference Proceedings*, 84, 895-912

Vannucci, M. & Colla, V. (2011). Novel classification method for sensitive problems and uneven datasets based on neural networks and fuzzy logic. *Applied Soft Computing*, 11(2), 2383-2390.

Wang, B., Mao, Z. & Huang, K. (2018). A prediction and outlier detection scheme of molten steel temperature in ladle furnace. *Chemical Engineering Research and Design*, 138, 229-247.

Wang, R., Li, H., Guerra, F., Cathcart, C. & Chattopadhyay, K. (2021). Development of quantitative indices and machine learning-based predictive models for SEN clogging. In: *AISTech 2021, proceedings of the iron & steel technology conference*, 1892-190.

Wu, X., Jin, H., Ye, X., Wang, J., Lei, Z., Liu, Y. & Guo, Y. (2021). Multiscale Convolutional and Recurrent Neural Network for Quality Prediction of Continuous Casting Slabs. *Processes*, 9(1), 33.

Yildirim, O., Baloglu, U. B., Tan, R. S., Ciaccio, E. J., Acharya, U. R. (2019) A new approach for arrhythmia classification using deep coded features and LSTM networks. *Computer methods and programs in biomedicine*, 176, 121-133.

