Improving steel making off-gas predictions by mixing classification and regression multi-modal multivariate models

Marcelo Magalhães do Carmo[1] **, Filipe W. Mutz**[1] **, Leandro C. Resendo**[1]

[1]Programa de Pós-graduação em Computação Aplicada
(PPComp IFES-Serra), Av. dos Sabiás, 330, 29166-630, Brazil

marcelo.mmcarmo@gmail.com,{filipe.mutz,leandro}@ifes.edu.br

Abstract. *This paper addresses the problem of real-time short-term multi-period off-gas prediction in a steel making batch process, denominated Linz-Donawitz Gas (LDG). Baselines, heuristic statistical methods, multi-modal multivariate Long Short-Term Memory (LSTM) and Ensemble Gradient Boosting Decision Tree (GBDT) strategies were proposed and compared. Proposed methods, mixing classification and regression tasks, achieved good results on recoverable LDG prediction, establishing a benchmark on subject for future works. Experiments suggest improvements from* 19.4% *to* 15.85% *on average in mean absolute percentage error (MAPE) over recent reviewed papers within a similar scenario at same steel making plant.*

1. Introduction

It is undeniable that steel is fundamental for the world development. Due to its relevance, steel production occupies a prominent position in the global economy. In an integrated steel making plant, there are many phases until the final product and, for competitively and efficiency, each step must be optimized and byproducts, generated in intermediate phases, must be harnessed. In the process to convert iron into a steel, molten iron and scrap are charged into a vessel called basic oxygen furnace (BOF) converter or just converter. Then, oxygen is blown with high pressure inside the converter to reduce carbon on total mass. This process produces an off-gas rich in carbon monoxide (CO) and carbon dioxide (CO_2). By the end of steel making process, the percentage of carbon in converted liquid steel will be lower than 1%, for commercial steel grades.

The decarburization process using top blowing oxygen on BOF is referred as Linz-Donawitz (LD) process, in honor to Linz and Donawitz cities in Austria, pioneers on this process. This way, the BOF is also known as LD converters [Fruehan 1998]. Oxygen blowing time vary typically from 15 to 30 minutes, while total conversion time vary from 30 to 40 minutes, and this process is commonly called **heat** or **batch** of liquid steel production [de Oliveira Junior et al. 2016]. The off-gas generated during oxygen blow is called Linz-Donawitz Gas (LDG). As this gas has a moderate calorific power due to high carbon monoxide (CO) concentration, it is commonly used as fuel for many subsequent processes on an integrated steel making plant, such as power plants. As a byproduct of steel making process, the LDG is correlated with production plan and activities.

LDG is an off-gas with variable chemical composition. It is considered recoverable when carbon monoxide (CO) content percentage reaches at least 25%, as typical value, [Fruehan 1998] and oxygen (O_2) below 2.5%, as illustrated in Figure 1. This figure indicates the percentage of main gases present on LDG (O_2, CO_2, CO e N_2) and its

variation over a typical oxygen blowing. At beginning and ending of blowing in BOF, the percentage content of CO is insufficient to classify the off-gas as recoverable. In general, the main recoverable LDG area is situated in the middle of heat time, highlighted between the vertical dotted lines.

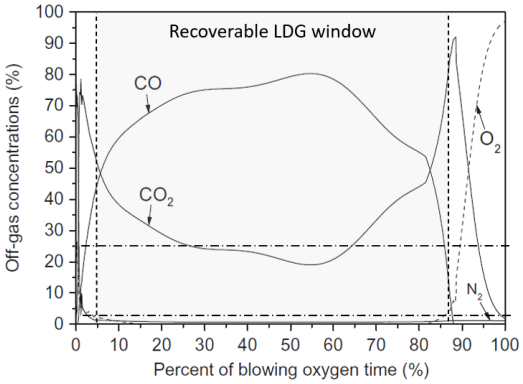

Figure 1. Typical LDG chemical concentration curve over time [Li et al. 2011]

According to works of Pena et al. [Pena et al. 2019], it was possible to predict steel making industry off-gas, including LDG, using statistical techniques. However, those auto-regressive techniques commonly fail to deliver good results on fast changing non-linear scenarios, which is characteristic of LDG batch generation. To minimize this impact, Pena et al. [Pena et al. 2019] proposed a post-processing heuristic over prediction of LDG using production data and statistical moving average techniques. In that investigation, three BOF in the same industrial plant of present work were analyzed, scoring 17.7%, 19.1% and 21.4% in mean absolute percentage error (MAPE) metric, and with average of 19.4% for all BOF. According to Colla et al. [Colla and et. al. 2019], similar results were obtained using recurrent neural networks for LDG prediction, scoring from 10% to 18% MAPE (10% on average for short-term forecasts), unfortunately lacking more robust benchmark numbers. At Wang et al. [Wang et al. 2020] paper, comparable relative metrics, like MAPE, for recoverable LDG results are missing.

In this paper, we propose 6 strategies, being 2 baseline benchmarks, 2 using multi-modal multivariate GBDT models and 2 using multi-modal multivariate LSTM neural network models, to predict multi period short-term generation of recoverable LDG. Multi-modal in this paper is composed by a temporal modal (multivariate sensor data) and a tabular modal (production system data), commonly present on any similar steel making production process. GBDT was chosen due to its performance when compared with model for tabular data or more complex deep learning models, as presented in [Gorishniy et al. 2021]. Recurrent neural networks strategies are frequently used for industrial scenarios involving time series [Colla and et. al. 2019, Wang et al. 2020, Zhao et al. 2018], being LSTM block one of most recent, flexible, and robust architecture for this purpose. The numerical results suggests that our strategies, to predict short-term generation of recoverable LDG, can establish a benchmark on literature.

We highlighted that the use of this multi-modal multivariate strategy in recoverable LDG generation prediction is the first contribution of this work. In addition, mixing classification and regression tasks improved results with a state-of-art architecture on sub-

ject, learning from time series and production data directly and simultaneously, reducing dependencies on experts' knowledge to build complex heuristics or more complex models. The second contribution of this paper is to establish a new benchmark on subject for future works.

The remainder of this paper is structured as follows: Section 2 describes used dataset, presenting an exploratory analysis; Section 3 presents model architectures, data transformation pipelines and hyper parameters configuration; Section 4 describes metrics; Experiments description and numerical results were presented in Section 5; finally, Section 6 provides some concluding remarks.

2. Exploratory analysis and variable selection

The investigation was performed on data from three BOF, that were labeled as 1, 2, and 3. The Figure 2 shows a typical production day from BOF 1. The first graph (time series) represents CO concentration on LDG (in %), the second graph (time series) represents the flag of recoverable LDG thresholds over time, and third (time series) represents BOF off-gas flow (in $^{kNm^3}/_{h}$). It is possible to observe that all series have characteristics of non-linearity over time and batch processing.

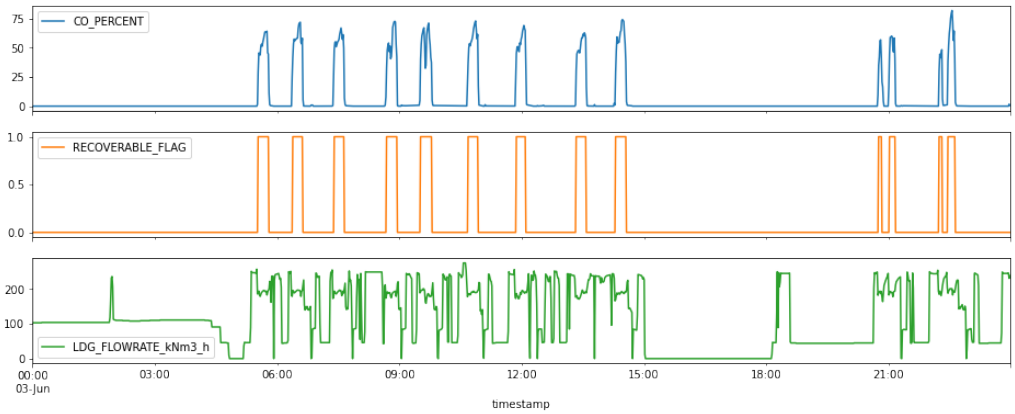

Figure 2. Typical day at BOF 1 production

2.1. Data Sources

In this work, each BOF data was collected from time series modal of LDG/BOF sensors and tabular modal from production system software. For time series data, we collected the main variables present on related works and added others from interviews with business process experts [Pena et al. 2019, Wang et al. 2020]. Time series were extracted with 1-minute frequency and aggregated by median values. For production data, the historical of heats were extracted from main production control systems. From heats on each BOF history, production plan and actual values were extracted. The join between temporal data and production data was made using the oxygen blowing actual start date. Over this collected time frame, many operational modes can be observed, such as different production rates, partial BOF maintenance and total production stoppage.

2.2. Production data analysis

From production tabular data, were extracted the chemical analysis information of molten iron (% of C, S, P, Mn, Si, etc.), percentage rates of scrap by heat, codes (or strategies)

of O_2 blowing, molten iron weight and BOF code number. The real data from production history were just used as predictors 10 minutes before the planned production start date time. Thus, it was considered the availability of those variables at this time because they are mandatory to start the production process on BOF, and usually they are available as soon as the production system collect the real information. Moreover, complementary data were computed with predictors, for instance, planned oxygen blowing start date (minutes), planned blowing duration time (minutes). These times were calculated based on production scheduling before heat stars. Therefore, the model will have all the simulated available actual data from production and planned process states at the current moment t.

2.3. Variable selection and generation of samples

Due to the large number of process variables and sensors (order of thousands), the use of every data available makes the problem almost impractical. Then, a pre-selection of variables was done by business knowledge and experiments.

Table 1. Summary of process selected tabular data and time series variables

Tabular variables per heat	**Units and comments**
Molten iron chemical analysis	% of C,Cr,Cu,Mn,P,S,Si,Ti in molten iron
Planned O_2 blowing total volume	kNm^3
Planned scrap rate	% of total weight
Actual scrap rate before O_2 blowing start	% of total weight or null if not available
Planned time duration of O_2 blowing	in minutes
O_2 blowing strategy (planned and actual)	code identifiers
Molten iron weight	in tons
O_2 blowing run-time counter	in minutes or null if O_2 blowing didn't started
Countdown timer for O_2 blowing start	in minutes (can be negative if delayed)
Time series variables	**Units and comments**
Gas chemical concentration analysis	% of CO,CO_2 and O_2 on BOF off-gas
BOF off-gas flow	in $^{kNm^3}/_h$
O2 lance activation flag	binary
O_2 blowing strategies (planned and actual)	code identifiers
O_2 blowing duration (planned and actual)	in minutes

Five main time series variables were used: BOF off-gas percentage concentration of CO, CO_2, O_2, off-gas flow ($^{kNm^3}/_h$), and a binary variable for O_2 lance activation flag. Historian of planned and actual O_2 blowing strategy codes were computed over time based on production data. Planned and actual blowing duration were computed over time as well. In summary, 9 time series and 36 tabular data variables, from production system were selected, as described in Table 1. Tabular data includes variables for next 2 heats in production schedule.

The strategy of sampling data follows a multi-modal pattern, where each sample has two groups of different data modal, defined as time series variables and process tabular data. Each training sample contain its own multi-modal data, including time series variables with the last 120 minutes values at time t, tabular process variables with next 2 scheduled heats at time t. Samples include answers for next 60 minutes of continuous LDG generation values, its binary quality threshold of recoverable LDG and real blowing start delays. Generation of samples were done by applying a rolling window strategy with

1-minute frequency. During this process, no scaling or learning tasks are applied to all data before train/validation and test split. Incomplete samples and outliers are discarded. For this work, based on business experience, samples with blowing start delays greater than 30 minutes are identified as outliers and excluded.

3. Methods

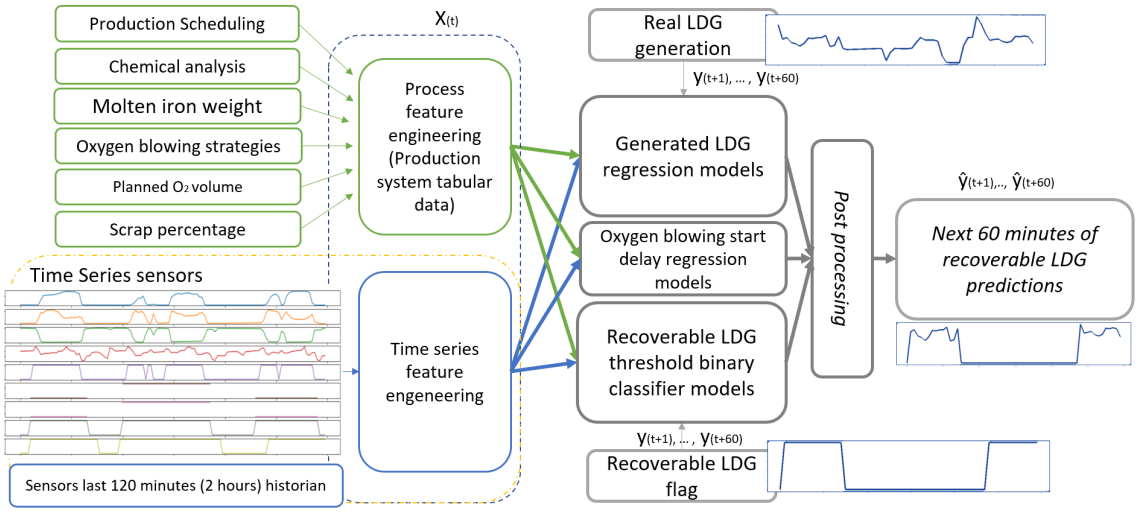

Figure 3. Model architecture for recoverable LDG prediction - *Ensemble GBDT*

The modeling proposed in this work compares an Ensemble Gradient Boosting Decision Tree (GBDT) [Friedman 2001, Ke et al. 2017] and a multi-modal multivariate recurrent neural network Long Short-Term Memory (LSTM) [Staudemeyer and Morris 2019]. Models were designed to accept as input samples of multi-modal data, like time counters, molten iron weight and chemical analysis, volume of oxygen blowing and time series sensors. Similar multi-modal design was proposed by Sala et al. [Sala et al. 2018], however applicable to a different steel making problem.

For GBDT architecture, data are inputted on transformation pipeline to feature extraction and aggregation process, creating the models' inputs, as illustrated in Figure 3. As the figure presents, tabular data indexed by time and heat variables (production sequences, chemical analysis, molten iron weight, oxygen blowing strategies, O_2 blowing programmed volumes and scrap percentage) are processed on feature engineering block and concatenated with data coming from time series feature engineering. The combination of these features generates X_t, which t is the current point in the time where model will predict the next 60 minutes of recoverable LDG. The predictions are denoted $\hat{y}_{t+i}, i \in \{1...60\}$. For each t, multiple models were trained in parallel for multiple strategies of prediction implemented in this work. Multi-period regressors are trained to predict BOF off-gas flow and multi-period binary classifiers to predict future threshold of recoverable LDG (where $CO > 25\%$ and $O_2 < 2,5\%$) for each $t + i$. Following the same idea presented in [Ke et al. 2017] for LightGBM, the GBDT implementation does not support multiple outputs on a single trained model, hence the strategy used was to train multiple models for multi-period prediction, with one model for each value predicted on $\hat{y}_{t+i}$.

The same basic architecture mechanism present in Figure 3 were adapted to predict recoverable LDG using recurrent neural networks. Nevertheless, eliminating the time

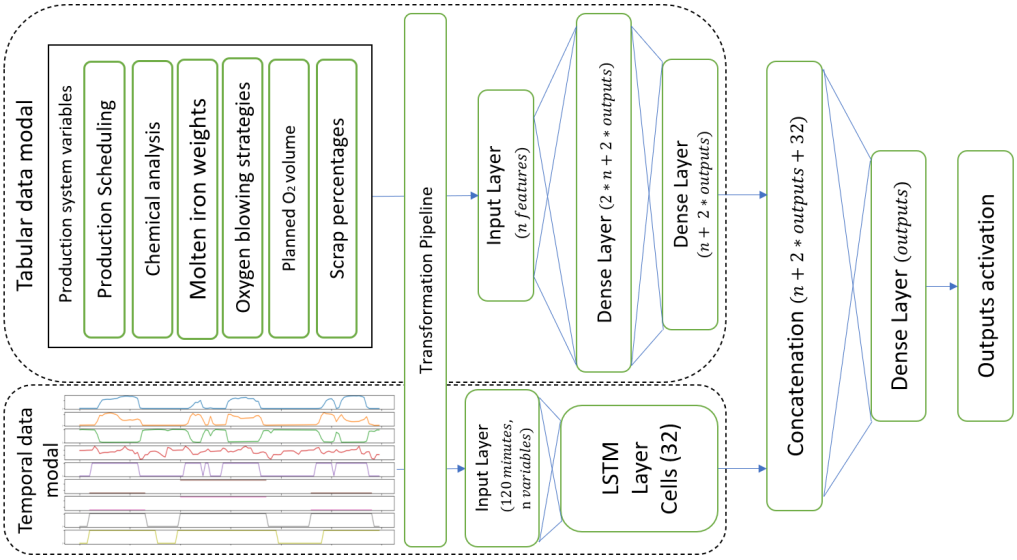

Figure 4. Multi-modal multivariate LSTM network architecture

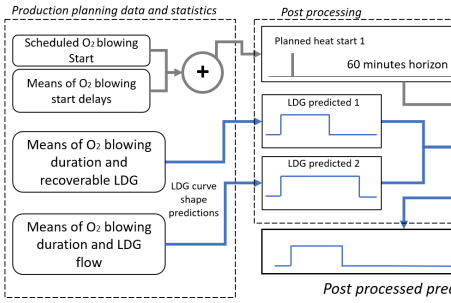

Figure 5. PROG-Baseline post processing strategy

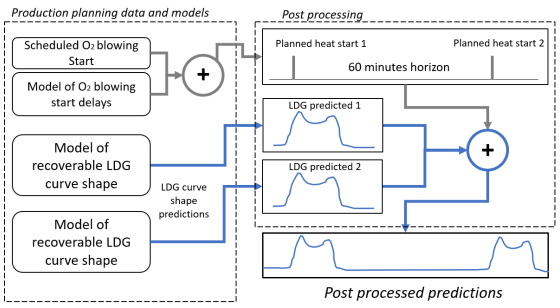

Figure 6. [GBDT/LSTM]-MM-PROG post processing strategy

series feature extraction and Ensemble GBDT by multi-modal multivariate LSTM with an appropriate pipeline. The architectural diagram of multi-modal multivariate LSTM models can be observed in Figure 4. In this figure, both tabular and time series data pass through a transformation pipeline, detailed in Section 3.1, to prepare the input data for neural network blocks. Input layers consider a specific neural network appropriated blocks for each modal, with different data shapes. For tabular data, multi-layer perceptron (MLP) architecture was chosen due to its simplicity and capability to be applicable as a sanity check model for tabular data [Gorishniy et al. 2021]. Two dense layers fully connected were used, with node numbers described on Figure 4 for each layer, by experiments. For time series data, LSTM cells were used. The concatenation block join exits from previous model layers and send to a fully connected final dense layer and output activation, according to prediction task. This strategy layout was based on Ofli et al. [Ofli et al. 2020] work, however adapting the modal blocks to multivariate time series and tabular data.

As illustrated in Figure 3, the post processing block strategy is applied to join trained model results and return final predictions. Six different post processing strategies were implemented and compared: 1st and 2nd are baseline strategies, named **ZERO-**

Baseline and **PROG-Baseline**, respectively; 3rd and 4th are based on trained GBDT strategies, named **GBDT-MM-PROG** and **GBDT-MM-MIX**; and, 5th and 6th are multi-modal multivariate LSTM trained strategies, named **LSTM-MM-PROG** and **LSTM-MM-MIX**.

ZERO-Baseline is a trivial method predicting zero values for all i in $\hat{y}_{t+i}$. It measures how unbalanced the recoverable LDG flag is for binary classification. If batches are sparse over time and delays are frequent, no prediction can be a good prediction. PROG-Baseline uses just steel making production data plan and statistics, using the scheduled O_2 blowing start time and duration for each planned heat in next 60 minutes, adjusting them by mean of O_2 blowing start delays (in minutes) and mean of recoverable LDG on training data. The returns of post processing predictions are the recoverable LDG, as illustrated in Figure 5. All next strategies should outperform these benchmarks.

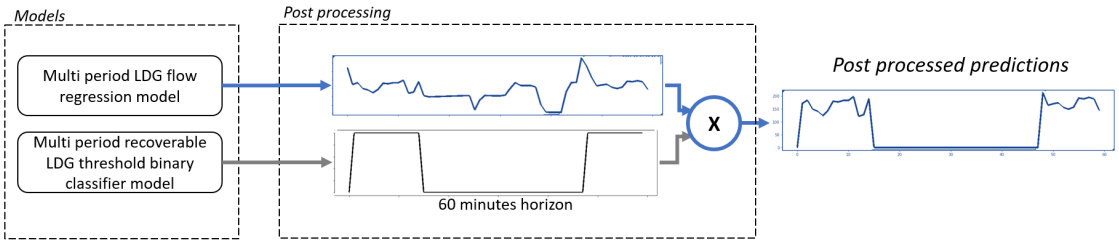

Figure 7. Post processing of mixed classifier and regressor strategies

GBDT-MM-PROG and LSTM-MM-PROG follows the same basic rules of post processing of PROG-Baseline, however adding trained model of delays in planned O_2 blowing start and exchange the statistical calculated means and duration by recoverable LDG curve prediction models, as presented in Figure 6. The reason to predict delays over planned O_2 blowing start time instead of predicting patterns of gaps between heats (time series model only approach) is related to the non-stationary production rate over time. Delays over time tends to be more stable and stationary, since industrial operators try, as much as possible, to execute production as planned.

GBDT-MM-MIX and LSTM-MM-MIX are the main strategies implemented in this work, using multi-output models to predict the continuous BOF off-gas flow and multi-output binary classifiers to predict recoverable LDG threshold over forecast horizon $\hat{y}_{t+i}$. For each $\hat{y}_{t+i}$, the regressor and binary classifier models' outputs are multiplied to mix the results, returning the final post processed prediction of recoverable LDG, as presented in Figure 7. From this simple strategy, delays, gaps, thresholds, and curve shapes are all inferred by model using multi modal data as input at once.

These strategies were motivated by non-linearity and batch process behavior, observed on recoverable LDG, as well as the need to multi-period prediction $\hat{y}_{t+i}$ for each X_t. On preliminary experiments, none of single regression model, GBDT or LSTM, without specific strategy for this batch process could straightforward achieve good results on recoverable LDG problem. This characteristic was also observed by correlated studies [Pena et al. 2019, Wang et al. 2020].

3.1. Data transformation pipeline and model hyper parameters

During the training task, the transformation pipeline learns statistical and scaling parameters, as well as feature extraction parameters when applicable. In both model strategies, ensemble or neural networks, a method to properly fill the missing data and outliers is performed. In these cases, historical average during recoverable LDG moments for each variable was filled. Real recoverable LDG values are scaled between 0 and 1. For evaluation and prediction tasks on test samples, the trained pipeline is applied.

Specifically for ensemble GBDT models, transformation pipeline extract features from time series. The feature chosen by experiments were statistical metrics of mean, standard deviation, median, energy and percentiles. In addition, the last 5 minutes of each time series were kept as features. For neural network models, the preprocessing pipeline for time series scales all features in datasets, converting the raw values to numbers between -1 and 1. Also, categorical data from tabular modal are binary encoded during the training, and categories that do not exist on training are ignored on prediction task.

For GBDT models, all hyper parameters were kept standard, as describe on *LightGBM* [Ke et al. 2017] library documentation used in this work (version $3.3.2$)[1]. It is aligned with the purpose of this paper, on establishing comparison baselines. For multi-modal multivariate LSTM model, the hyper parameters of number of nodes on layers were described on each block in Figure 4. All other parameters are present on Table 2, and follow typical values present on *TensorFlow/Keras* [Abadi et al. 2015] library documentation (version 2.6)[2] or were chosen by experiments.

Table 2. Hyper parameters for multi-modal multivariate LSTM models

Blocks	**Layers**	**Numbers**	**Activation**	***Dropout***
Time series modal block	LSTM	32	*tanh*	0.2
Tabular modal block	Dense	2	*relu*	0.2
Regression output block	Dense	1	*linear*	-
Binary Classifier ouput block	Dense	1	*sigmoid*	-
Optimizer	**Early Stop**	**Patience**	**Restore best values**	
Adam	*val_loss*	20 epochs	Yes	
Train tasks	**Epochs**	**Batch**	**Learning rate**	
1st - partial fit	10	2000	0.001	
2nd - final fit	1000	200	0.0005	
More hyper parameters	**Loss function**		**Validation split**	
Regression	Mean Square Error		0.2	
Classifier	Binary Cross Entropy		0.2	

4. Metrics

The performance metrics chosen were root mean square error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE). **RMSE** (Equation 2) and **MAE** (Equation 3) give an idea of the magnitude of error in $^{kNm^3}/_{h}$, while **MAPE** (Equation 1) give an idea of relative error over different BOF LDG predictions. All chosen metrics are commonly used in comparable problems and general forecasting problems [Colla and et. al. 2019, Luca Avila and De Bona 2020, Pena et al. 2019,

[1]https://lightgbm.readthedocs.io/en/v3.3.2/index.html

[2]https://www.tensorflow.org/versions/r2.6/api_docs/python/tf

Zhao et al. 2018]. This work also introduced the use of a different metrics for this type of problem, **Precision** and **Accuracy** (Equations 4 and 5). They are commonly used in classification problems, but as we mixed regression and binary classification models, the correct indication of when LDG will be recoverable is so important as its absolute value prediction. Dealing with MAPE singularity issue, the metric $\mathbf{MAPE}_{adj}$ were defined (Equation 6) and will replace MAPE scores on results. Similar adjustment was made in comparable paper [Pena et al. 2019].

$$MAPE = \frac{100}{n}\left(\sum_{t=1}^{n}\left|\frac{y_t - \hat{y}_t}{y_t}\right|\right) \quad (1)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2} \quad (2)$$

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|y_t - \hat{y}_t| \quad (3)$$

$$\text{Precision} = \frac{\sum \text{True positive}}{\sum \text{True positive} + \sum \text{False positive}} \quad (4)$$

$$\text{Accuracy} = \frac{\sum \text{True predict}}{\sum \text{Predict}} \quad (5)$$

$$MAPE_{adj} = \begin{cases} MAPE(y_t, \hat{y}_t), & (|y_t| > 1) \\ 100, & (|y_t| \leq 1) \wedge (|\hat{y}_t| > 1) \\ 0, & (|y_t| \leq 1) \wedge (|\hat{y}_t| \leq 1) \end{cases} \quad (6)$$

5. Experiments and Results

The experiments were accomplished for each BOF independently. Unfortunately, due to the specific equipment and sensors characteristics of each BOF, it was not possible in this work to generalize a model to all converters. For training and validation, this work used 279059, 325434 and 335071 multi-modal samples from BOFs 1, 2, and 3, respectively. We used 80% of samples in the training and 20% on validation. For test were used 83776, 87139 and 106383 samples from BOF 1, 2 and 3, respectively. It must be highlighted that test samples were not used in any training or validation tasks, and they were sampled after dates of train/validation datasets. Samples are defined as described in Section 2.3

Table 3. Table of results by BOF and model prediction strategy

Model	BOF	$\boldsymbol{MAPE}_{adj}$(↓)	RMSE(↓)	MAE(↓)	σMAE(↓)	Accuracy(↑)	Precision(↑)
ZERO-Baseline	BOF1	27.33	103.34	53.63	0.48	0.727	0
PROG-Baseline	BOF1	22.48	92.52	45.93	3.43	0.803	0.606
LSTM-MM-PROG	BOF1	16.94	79.08	33.48	8.74	0.838	0.709
GBDT-MM-PROG	BOF1	16.86	78.52	33.38	9.51	0.840	0.715
LSTM-MM-MIX	BOF1	15.62	74.61	30.97	9.10	0.857	0.746
GBDT-MM-MIX	BOF1	**15.15**	73.44	30.07	9.44	**0.861**	0.759
ZERO-Baseline	BOF2	27.86	106.57	54.93	0.70	0.721	0
PROG-Baseline	BOF2	24.23	86.87	45.51	3.44	0.799	0.605
LSTM-MM-PROG	BOF2	19.79	79.14	37.44	9.68	0.834	0.705
GBDT-MM-PROG	BOF2	19.53	78.62	37.06	10.14	0.836	0.711
LSTM-MM-MIX	BOF2	16.82	70.50	31.70	9.01	0.865	0.766
GBDT-MM-MIX	BOF2	**16.34**	69.29	30.77	9.40	**0.869**	0.780
PROG-Baseline	BOF3	25.31	66.73	36.09	2.08	0.795	0.556
ZERO-Baseline	BOF3	24.68	69.21	32.58	0.40	0.753	0
GBDT-MM-PROG	BOF3	17.78	52.99	23.64	5.83	0.852	0.695
LSTM-MM-PROG	BOF3	17.76	52.67	23.47	5.60	0.850	0.693
LSTM-MM-MIX	BOF3	16.68	51.20	22.30	6.11	0.863	0.734
GBDT-MM-MIX	BOF3	**16.06**	49.93	21.41	5.83	**0.870**	0.748
GBDT-MM-MIX (best)	All	**15.85**	64.22	27.42	8.22	**0.867**	0.762

Table 4. Table of results on MAPE$_{adj}$ over prediction time horizon

Model	BOF	$\hat{Y}_{(t+1 \dots t+10)}$	$\hat{Y}_{(t+11 \dots t+20)}$	$\hat{Y}_{(t+21 \dots t+30)}$	$\hat{Y}_{(t+31 \dots t+40)}$	$\hat{Y}_{(t+41 \dots t+50)}$	$\hat{Y}_{(t+51 \dots t+60)}$
PROG-Baseline	BOF1	19.34	21.60	23.64	24.13	23.67	22.51
LSTM-MM-PRO	BOF1	8.84	13.91	18.85	20.22	20.07	19.73
GBDT-MM-PROG	BOF1	7.96	13.95	18.96	20.27	20.15	19.86
LSTM-MM-MIX	BOF1	6.88	12.99	17.17	18.89	19.12	18.69
GBDT-MM-MIX	BOF1	6.05	12.73	16.69	18.65	18.49	18.27
PROG-Baseline	BOF2	21.09	23.93	25.77	25.90	25.11	23.57
LSTM-MM-PRO	BOF2	10.51	16.90	22.24	23.58	23.17	22.35
GBDT-MM-PROG	BOF2	9.85	16.45	21.95	23.47	23.11	22.35
LSTM-MM-MIX	BOF2	7.93	14.03	18.34	20.25	20.28	20.09
GBDT-MM-MIX	BOF2	7.24	13.13	17.71	20.23	20.09	19.67
PROG-Baseline	BOF3	23.46	26.14	26.20	26.52	25.72	23.84
LSTM-MM-PRO	BOF3	9.82	15.91	20.13	20.52	20.33	19.81
GBDT-MM-PROG	BOF3	9.57	15.85	20.20	20.67	20.47	19.91
LSTM-MM-MIX	BOF3	8.00	14.38	18.76	19.69	19.86	19.37
GBDT-MM-MIX	BOF3	7.63	13.60	17.96	19.19	19.08	18.91
GBDT-MM-MIX (best)	All	6.97	13.15	17.45	19.36	19.22	18.95

Table 3 presents an overview of results, grouped by BOF and sorted by metric MAPE$_{adj}$. Standard deviation of MAE (σMAE) was also included. All proposed strategies outperform ZERO-Baseline and PROG-Baseline for all BOF. Highlighted on BOF3, ZERO-Baseline MAPE$_{adj}$ outperform PROG-Baseline, due to higher delays over planning. All MIX strategies, on MAPE$_{adj}$, outperform it's counterpart PROG strategy by at least 1% point, and GBDT-MM-MIX outperform PROG-Baseline by at least 7% points. GBDT-MM-MIX also improved Accuracy and Precision over all models, increasing the reliability of predictions. In the end of Table 3, it's included the best model with the average of its metrics for all BOF.

Table 4 present results of MAPE$_{adj}$ aggregated by 10-minute predictions. It is possible to observe, highlighted at end, that almost all strategies first 30-minutes have better scores than last 30-minutes. Even though models have worse performance on last 30-minutes, they outperform PROG-Baseline strategy, which is the default information provided by production planning system from any BOF.

The Figure 8 shows four frame examples in rolling time window on every 5 minutes. PROG-Baseline, LSTM-MM-MIX and GBDT-MM-MIX predictions are compared with ground truth values (Real Recoverable LDG). These 4 frames also show the evolution of predictions over time. On 1st frame, PROG-Baseline predicted starts for next 2 heats, but real values are delayed by 3 and 5 minutes respectively (last delay visible on 2nd frame). At 2nd frame, model predictions were very close to curve shape and actual values, but both missing 1 minute at end of first heat and 1 minute at start of second. On 3rd and 4th frames, first heat is finishing and the second is planned on last 30-minutes. Both models predicted reasonably well the recoverable LDG start and end, but couldn't predict the shape of the curve, averaging the predictions over time, exemplifying worse performance on Table 4 for last 30-minutes.

As presented results in Table 3, proposed models succeeded in predict the multi period recoverable LDG using multi-modal multivariate mixed strategies. Result suggests an improvement compared to Pena et al. [Pena et al. 2019] work at same industrial plant. Analyzing the prediction horizon 4, the scores obtained by best strategy on average

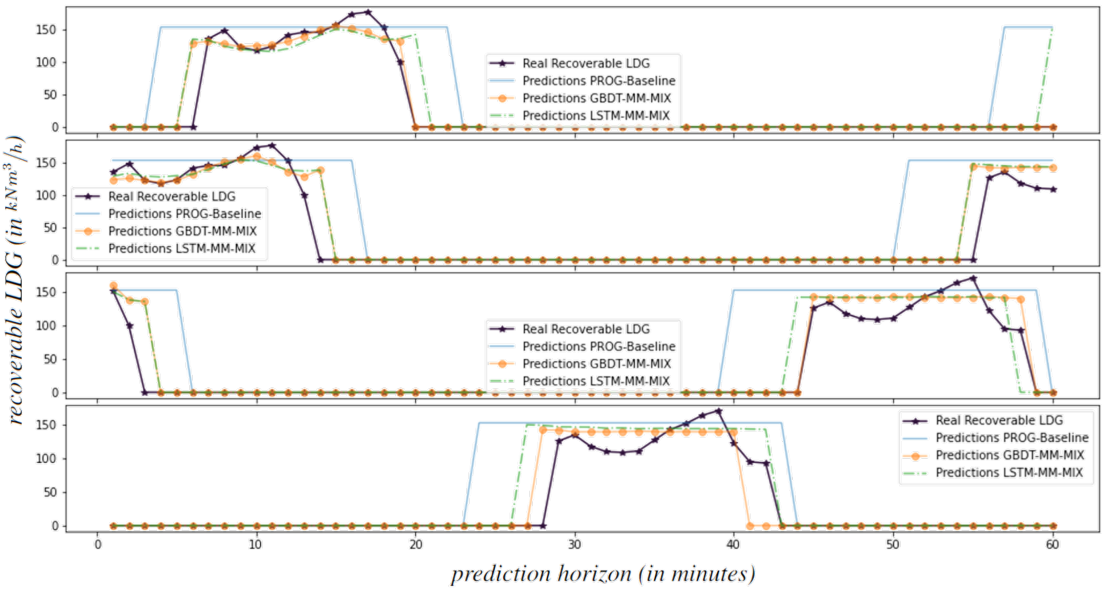

Figure 8. Four samples of predictions in rolling time window on test dataset

were between 6.97% and 19.36%, which are better for very short-term predictions when compared to results in Colla et al. [Colla and et. al. 2019] (6.97% against best 10%).

6. Conclusions

The objective of this paper is to compare and improve strategies for recoverable LDG prediction found in the literature. The mixed strategy proposed in this work suggests, by experiments, that both multi-modal multivariate LSTM and GBDT has potential to predict recoverable LDG generation over time. Besides the application presented, the architectures proposed can be adapted to any industrial process by batch with a time series predictions.

Beyond just quantitative metrics, each model has its own qualitative advantages and disadvantages. GBDT on open-source available libraries, for instance LightGBM used, could not train multi output in a single model and demands a feature engineering pipeline over time series. Recurrent neural networks are more flexible architecture, being able to be trained for multiple outputs, multi-modal multivariate data and learning to extract features on time series. Moreover, the LSTM proposed strategy achieved second best results systematically on test datasets.

Finally, no hyper parameter optimization or more complex deep neural networks were exhaustively explored. Future works can further improve the contributions presented by focusing efforts on hyper parameters optimization, improve feature extraction from time series, review new proposals of GBDT for multi output training or compare results with new Transformer neural architectures.

Acknowledgments

This research was supported by ArcelorMittal Brasil S.A. steel company, and by FAPES and CAPES (Grant nº: 2021-2S6CD, nº FAPES 132/2021) through the PDPG (Programa de Desenvolvimento da Pos-Graduação - Parcerias Estrategicas nos Estados).

References

[Abadi et al. 2015] Abadi, M. et al. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

[Colla and et. al. 2019] Colla, V. and et. al. (2019). Assessing the efficiency of the off-gas network management in integrated steelworks. *Materiaux & Techniques*, 107(1):104.

[de Oliveira Junior et al. 2016] de Oliveira Junior, V. B., Pena, J. G. C., and Salles, J. L. F. (2016). An improved plant-wide multiperiod optimization model of a byproduct gas supply system in the iron and steel-making process. *Applied energy*, 164:462–474.

[Friedman 2001] Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

[Fruehan 1998] Fruehan, R. J. (1998). *The making, shaping and treating of steel: steelmaking and refining volume*. AISE steel Foundation.

[Gorishniy et al. 2021] Gorishniy, Y., Rubachev, I., Khrulkov, V., and Babenko, A. (2021). Revisiting deep learning models for tabular data. *Advances in Neural Information Processing Systems*, 34.

[Ke et al. 2017] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30:3146–3154.

[Li et al. 2011] Li, S., Wei, X., and Yu, L. (2011). Numerical simulation of off-gas formation during top-blown oxygen converter steelmaking. *Fuel*, 90(4):1350–1360.

[Luca Avila and De Bona 2020] Luca Avila, R. d. and De Bona, G. (2020). Financial time series forecasting via ceemdan-lstm with exogenous features. In *Brazilian Conference on Intelligent Systems*, pages 558–572. Springer.

[Ofli et al. 2020] Ofli, F., Alam, F., and Imran, M. (2020). Analysis of social media data using multimodal deep learning for disaster response. *arXiv preprint arXiv:2004.11838*.

[Pena et al. 2019] Pena, J. G. C., de Oliveira Junior, V. B., and Salles, J. L. F. (2019). Optimal scheduling of a by-product gas supply system in the iron-and steel-making process under uncertainties. *Computers & Chemical Engineering*, 125:351–364.

[Sala et al. 2018] Sala, D. A., Jalalvand, A., Van Yperen-De Deyne, A., and Mannens, E. (2018). Multivariate time series for data-driven endpoint prediction in the basic oxygen furnace. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1419–1426. IEEE.

[Staudemeyer and Morris 2019] Staudemeyer, R. C. and Morris, E. R. (2019). Understanding lstm–a tutorial into long short-term memory recurrent neural networks. *arXiv preprint arXiv:1909.09586*.

[Wang et al. 2020] Wang, T., Leung, H., Zhao, J., and Wang, W. (2020). Multiseries featural lstm for partial periodic time-series prediction: A case study for steel industry. *IEEE Transactions on Instrumentation and Measurement*, 69(9):5994–6003.

[Zhao et al. 2018] Zhao, J., Wang, W., and Sheng, C. (2018). Industrial time series prediction. In *Data-Driven Prediction for Industrial Processes and Their Applications*, pages 53–119. Springer.