SOFT SENSOR: MACHINE LEARNING TRADICIONAL OU DEEP LEARNING?*

Pablo Drumond[1]
Daniele Kappes[2]
Cassio de Moraes[3]
Eduardo Oliveira[4]
Mariana Teixeira[5]

Resumo
Para atingir a qualidade desejada no concentrado da flotação, numa usina de mineração de ferro, o operador tomava decisões com latência mínima de 2 horas, o equivalente ao tempo para recebimento de resultados da análise laboratorial. Um tempo morto tão elevado, tornava difícil colocar a planta em regime quando ocorria uma variação na alimentação da flotação ou na qualidade desejada. Dessa forma, o objetivo do presente trabalho foi construir um sensor virtual capaz de estimar o teor SiO2 e permitir a atuação do operador em um período mais curto. A alta variabilidade na alimentação, adicionada às dificuldades inerentes da predição envolvendo séries temporais tornaram o projeto singular. Foram aplicadas técnicas de mineração de dados e seleção de features em uma base de 03 meses abrangendo variáveis da Flotação e também da Deslamagem, e obtiveram-se entradas para elaboração de 03 modelos de Machine Learning: Random Forest, GradientBoostedTrees e MultiLayerPerceptron. O Soft Sensor baseado em redes neurais artificiais se mostrou estatisticamente mais eficiente na etapa de teste. Na versão online, o baixo erro médio absoluto obtido comprovou a robustez do modelo, entregou agilidade para a operação e certificou o poder dessa abordagem em processos industriais com alta latência de resultados laboratoriais.
Palavras-chave: Flotação; Machine Learning; Soft Sensor; Multi-Layer Perceptron

SOFT SENSOR: TRADITIONAL MACHINE LEARNING OR DEEP LEARNING?

Abstract
In an iron ore mining plant, in order to achieve the desired quality in the flotation concentrate, the operator took decisions based on the result of a laboratory analysis that took 2 hours minimum to be available. Such a long dead time made it difficult to put the plant in steady state when a variation in the feeding of the flotation occurred or when the output specification changes. Thus, the aim of the present work is to report a construction of a virtual sensor capable of estimating the SiO2 content and that allowed the operator to perform a responsive action in a shorter period. The high variability in feeding on top of the inherent difficulties in the prediction involving time-series made the project unique. Data mining techniques and feature selection were applied on a 03 months database covering Flotation as well as the Desliming variables, and different datasets were used in order to obtain the following three Machine Learning models: Random Forest, Gradient Boosted Trees and Multi-Layer Perceptron. The Soft Sensor based on artificial neural networks was more efficient in the test stage. In the online version, the low average absolute error obtained proved the robustness of the model, delivered agility to the operation and confirmed the power of this approach in industrial processes with high latency of laboratory results.
Keywords: Froth Flotation; Machine Learning; Soft Sensor; Multi-Layer Perceptron.

[1] *Bachelor of Engineering (BEng) focused in Control and Automation from Federal University of OuroPreto, Graduated in Industrial Engineering and Data Science & Big Data from PUC-MG, Data Engineer, Data Science, IHM Stefanini, Belo Horizonte, Minas Gerais, Brazil.*

[2] *Bachelor of Engineering (BEng) focused in Control and Automation from Federal University of Minas Gerais, Data Scientist, Data Science, IHM Stefanini, Belo Horizonte, Minas Gerais, Brazil.*
[3] *Bachelor of Engineering (BEng) focused in Mechatronics from Federal Center of Technological Education of Minas Gerais, Data Scientist, Data Science, IHM Stefanini, Belo Horizonte, Minas Gerais, Brazil.*
[4] *Master of Science in Electrical Engineering from Federal University of Minas Gerais, Leader of Data Science and AI, Data Science, IHM Stefanini, Belo Horizonte, Minas Gerais, Brazil.*
[5] *Bachelor of Statistics from Federal University of Minas Gerais.*

1 INTRODUCTION

Froth Flotation is a mineral separation process based on two particles surface characteristics: hydrophilicity and hydrophobicity. Hydrophobic particles adhere to the air bubbles surface, have their density changed, then ascend and overflow the cell, creating a froth, which is constantly removed from the cells. Hydrophilic particles remain in the water, sediment and create an underflow with a higher concentration of Fe, which flows into the next cell in order to refine the same process. Since some Fe is carried outward along with the froth a closed circuit is needed to increase the metallic recovery. This process is one of the most effective ways to take SiO_2 out of the slurry and therefore increase the concentration of iron ore fines in the product stream. Figure 1 shows a froth flotation cell.

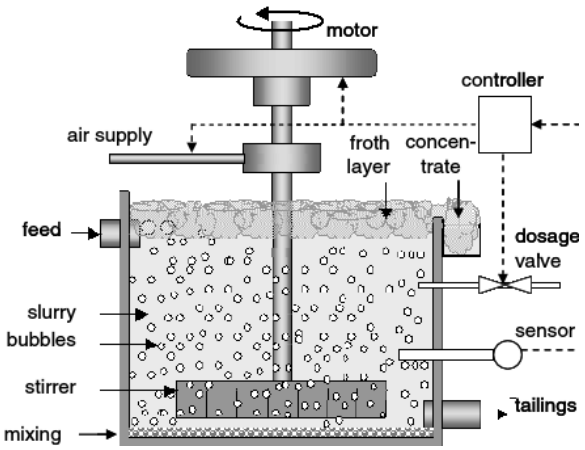

Figure 1 - Froth flotation cell with sensor, actuators and control system.
Source: https://www.researchgate.net [1]

In a typical froth flotation plant, the operator takes some actions, such as increasing the reagents flow or reducing feed rate, in order to achieve the desired quality. In the production facility studied, those actions can only have their effectiveness confirmed by the laboratory analysis results, within 02 hours. Such a long dead time made it difficult to put the plant in a steady state when feeding quality varies or when the output specification changes.

Therefore, a possible solution for this issue, that will be referenced as SiO_2 Prediction Problem, would be a predictive model capable of estimating the concentration of SiO_2 in the product stream, which turned into the central objective of the project. The main reason to support the investment is related to product quality, i. e., meeting the product specification with lower resource consumption, shorter time and thus delivering a better control of the plant with lower cost as opposed to the costly online slurry analyzers.

In broad terms, applying prediction in mineral processing, especially Froth Flotation, is not a new concept. There are several types of research involving different machine learning techniques, and it is worth mentioning the following as the being the ones most relevant to this work:

- Shabahzi et al. [2] investigated the effect of process variables on the flotation rate and recovery using Random Forest;
- Nakhaei and Irannajad [3] used various neural networks architectures in order to predict the metallurgical performance of the flotation column in a copper plant;
- Despotoviç [4] built a Support Vector Regression model to estimate deinking flotation performance in the paper recycling process.

Nevertheless, the majority of the previous works applied just train, validation and test steps in order to prove the concept and they did not report deploying them nor achieving online phase.

Among the challenges that made this paper pertinent, it is possible to mention the non-existence of machine vision system, thus features like bubble size and froth speed were not available to take part as model inputs. Furthermore, on account of being essentially a dynamic process, there were no strong correlations between variables and output, and traditional statistical methods were not efficient enough to describe the relationship among variables, which required great efforts in feature engineering phase.

2 METHODOLOGY

According to Mariscal et al. [5], Cross-Industry Standard Process for Data Mining, known as CRISP-DM, is the most widely used form of data-mining model, and it was also assured by KDnuggets research [6] to be the top methodology for data science projects. Figure 2 shows the cycle and phases of the process.

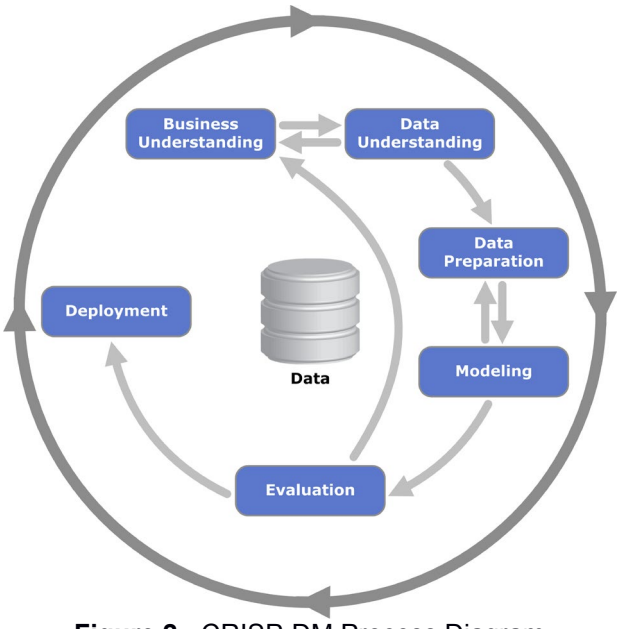

Figure 2 - CRISP-DM Process Diagram
Source: https://www.kdnuggets.com

Despite having no enhancements since its release, even as big data advanced, it proved to be a powerful approach for analytics and consequently was selected to guide the development of the SiO2 prediction model.

It is important to mention that even though CRISP-DM proposes a cycling approach, which means the developing model is enhanced within each loop, the task behind its phases will be described in this paper as a single thread.

2.1 Business Understanding

A preliminary assessment on the product stream showed that variability of SiO_2 over a relatively short period of time was very common, which indicated recurrent operator attempts to meet product specification and confirmed the hypothesis that the flotation plant is indeeddifficult to operate. Figure 3 shows that variability.

Figure 3 - Concentration of SiO2 in the product stream
Source: Author's own figure

Contrary to the initial belief, the objective is not to minimize SiO_2 in the output stream. In fact, blending iron ore and ending up with a product within the admissible range is the major goal of a mineral processing plant. Strictly speaking, sometimes a poor slurry delivered by flotation is advantageous to the business.

Thus, after the initial assessment, the goals were established as being: to find the root causes of SiO_2 variance; to understand the influence of certain variables such as surfactants and wetting agents flow, and then, to build a model taking into account the most important features for SiO_2 estimation.

2.2 Data Understanding

From the start of the analysis, the PIMS showed itself to be the appropriate long-term source for retrieving necessary data for modeling. It stores and aggregates not just time series of process variables but also sampling results which are published by the LIMS.

The initial dataset was 03 months long. At the first cycle, a primary concern was to assure that any changes in the flotation circuit had not happened in the previously mentioned period. It could occur mainly because most of the time a cell designed to be the second stage of Scavenger is turned into a third Cleaner stage although without any notification in the automation system.

In order to understand the plant performance, it would be interesting to include in the data collection a high variance of ROM[1]. On top of that, there was another assumption related to the variable extraction: the value must be available online, which means that during runtime the model must have access to the current value of the variable considered as input.

Finally, after meeting the above requirements, data were extracted and loaded into a relational database in order to make it easy operations like select, filter and join.

Furthermore, it is important to highlight the sampling operation, by which the concentration of SiO2 is determined in the flotation and deliming processes. The lab sample, which is drawn from the process every 2 hours, consists of small partitions of slurry that are collected at 12-minute intervals. Therefore, one lab result reflects only a combination of those partitions and not the process scenario at a specific moment. The mentioned sampling method may lead to considerable loss of information since some process changes cannot be properly tracked during a 2-hour time frame.

2.3 Data Preparation

The dataset shape was 700k rows and 120 columns, which corresponds to 10 seconds sampling of the process and lab variables. The number of rows was good enough to avoid underfitting, nevertheless, the number of columns could impact in generalization performance, which required extra efforts for selection and reduction.

Data preparation or pre-processing handles and transform raw data so that knowledge could be easily discovered [7]. The following enrichment methods were performed:

- Purging plant shutdown periods based on feeding rate value;
- Filtering outliers with z-score and threshold adjustment;
- Deleting features with extreme low variance, for instance, some cell levels.

The correlation coefficient between variables and target was extremely weak, which confirmed the initial hypothesis inherent to the dynamic process. However, it was worth reducing some redundant features, for instance, densities and motor currents and air insufflation rate among the cells.

The first dataset (DS#1) was prepared after applying Feature Ranking with Recursive Feature Elimination, known as RFE. For this purpose, it was used a Support Vector Regression algorithm as an external estimator in order to assign weights to each feature. Then, the process variables were recursively pruned until the desired number was reached, 20.

---

1 Run of Mine

The second dataset (DS#2) was prepared after applying Least Absolute Shrinkage and Selection Operator. Lasso is a regression analysis method that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the statistical model it produces [8]. This method selected 41 variables using 0,025 as the regularization term.

It is relevant to note that, despite similar principles, the methods above elected quite different features. Then, it was decided to prepare a third dataset (DS#3) made up of common variables between the output of RFE with 40 features and Lasso with 41 features. So it was selected 26 process variables.

Although PCA is a notorious tool for dimension reduction, it was not carried out in this study due to a loss of understanding of flotation process. In other words, it would be difficult to interpret whether the transformed vectors were clear enough according to the process knowledge.

2.4 Modelling

Regardless of machine learning technique to be implemented, the approach would rely on three phases: training, validation and test. Each step counting on a data subset from each the three major datasets would be prepared by the feature engineering, split as follows:

- Test subset: last 12 days of the working dataset;
- Train subset: 80% of the remaining rows, randomly selected;
- Validation subset: the remaining rows.

In general, in the first step the model parameters are adjusted using an optimization algorithm which aims to minimize error measurement, using the difference between estimated and expected output, also called supervised learning technique.

The second step is applied to tune hyperparameters in order to overcome bias or overfitting behaviors. It is followed by a comparison between the results of each attempt and a selection of which best fits the data. Alternatively, it is possible to merge Train and Validation subsets in order to use cross-validation technique then, choosing the optimal weights among models trained against K-Folds.

Lastly, the performance of the trained model is assessed with unseen data, which simulates a real context with a smaller dataset that follows the same probability distribution of the training phase.

Gradient Boosted Tree, Random Forest and Multi-Layer Perceptron were chosen to try to solve SiO_2 Prediction Problem. Root Causes Analysis was carried out to better understand the influence of some variables on the chemical concentration of the output stream.

Froth Flotation is a dynamical process, which means that training the model with the variables at the same instant of time would certainly result in a poor performance. In other words, SiO2 at a specified timestamp is not related to the reagents flow in the

same timestamp, which was the reason one of the biggest challenges was to find out the right lag between variables and the target.

2.4.1 Gradient Boosted Tree

Gradient Boosting is a machine learning technique that can be used for regression and classification problems. It produces a prediction model in the form of an ensemble of simple estimators, typically decision trees. In this method, the learning procedure consecutively fits a new model to provide a more accurate estimate of response variable [9].

2.4.2 Random Forest

Random Forest is a machine learning technique that can also be used for regression and classification problems. It constructs a multitude of decision trees at training time and outputting the mean prediction (regression) of the individual trees. “It consists of a chosen number of decision trees. Each of the decision tree models is learned on a different set of rows (records) and a different set of columns (describing attributes), whereby the latter can also be a bit-vector or byte-vector descriptor (e.g. molecular fingerprint). The row sets for each decision tree are created by bootstrapping and have the same size as the original input table. For each node of a decision tree, a new set of attributes is determined by taking a random sample of size sqrt(m) where m is the total number of attributes. The output model describes a random forest and is applied in the corresponding predictor node using a simple majority vote” [10].

2.4.3 Root Cause Analysis

Another approach used in this study was to try to build a root cause analysis hoping to get the causes of the SiO_2 percentage being out of specification. Two different approaches were tried : (1) using process and quality variables without any lag and (2) using this same dataset but applying a one-hour lag to the process variables due to the silica’s residence time. Then, after deciding for the second approach, it was necessary to do some data manipulation before applying the Decision Tree algorithm to get a root cause analysis: it was classified with the label “meet the requirement” those data which the silica percentage was between 2.9% and 3.9% and as “does not meet the requirement” those which the silica percentage was less than 2.9% and greater than 3.9%.

In the next step, the dataset was balanced, so data were filtered, leaving the same amount of data classified as “meet the requirement” and classified as “does not meet the requirement”. Then, the dataset was split in training set (86%) and test set (14%) and it was applied the Decision Tree Algorithm to both datasets.

2.4.4 Multi-Layer Perceptron

The Perceptron was introduced by McCulloch and Pitts as a computational version of a biological neuron, which can be understood as a weighted sum of the inputs used in an activation function to compute an output, as shown in Figure 4.

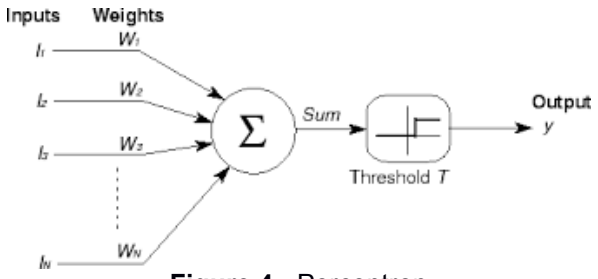

Figure 4 - Perceptron
Source: http://wwwold.ece.utep.edu/research/webfuzzy/docs/kk-thesis

Based on that abstraction, Minsky and Papert proposed perceptrons organized in layers interconnected and confirmed Neural Networks with more than one hidden layer as a powerful tool capable of solving nonlinear separation problems.
As it turned out, the network with only one hidden layer achieved a low performance, R-squared was about 0,59 in the training phase. By using deep learning techniques along with a graphical analysis of cost function, the topology dramatically changed, and it allowed to improve R-squared to 0,95 in the same phase. Which seemed to be promising, in fact, exposed an overfitting behavior when the test phase resulted in R-squared about 0,2. Then, in order to address this issue as well as to avoid getting stuck in a local minimum, the following actions were performed during the training:

- Adjusting regularization term: adds penalties to the coefficients;
- Applying drop-out: ignores a random fraction of nodes;
- Adjusting decaying rate: drops the learning rate as the training proceeds;
- Setting batch size: defines the number of samples considered in each batch.

Finally, the topology used to solve SiO_2 prediction problem counted on 08 hidden layers and up to 50 nodes in one of them. It used Rectified Linear Unit (ReLU) as activation function and Backpropagation with Adam as the training algorithm.
The table below shows the process variables of DS#1, used as inputs in the network:

Table *1* – DS#1 Features

Inputs	
Conditioning Tank - pH	CF-05 Cell (Rougher) - Amine Flow
CF-01 Flotation Cell (Rougher) - pH	Conditioning Tank - Starch Flow
Conditioning Tank - Density	CF-03 Cell (Rougher) Level
Grinding Feeder - Current	CF-05 Cell (Rougher) Level
Recirculation Pump - Current	CF-13 Cell (Cleaner 2) Level
Recirculation Pump - Speed	CF-21 Cell (Scavenger 2) Level
CF-05 Cell (Rougher) - Stirrer Current	SiO_2 - Output Stream (02h Lag)
CF-07 Cell (Cleaner 1) - Stirrer Current	Al_2O_3 - Output Stream (02h Lag)
CF-11 Cell (Cleaner 2) - Stirrer Current	Fe- Output Stream (02h Lag)
CF-15 Cell (Scavenger 1) - Stirrer Current	Mn - Output Stream (02h Lag)
CF-17 Cell (Scavenger 1) - Stirrer Current	P - Output Stream (02h Lag)
CF-19 Cell (Cleaner 3) - Stirrer Current	Conditioning Tank - Amine Flow
CF-09 Cell (Cleaner 1) - Amine Flow	

3 RESULTS

3.1 Evaluation

The next topics show the best result using the test subset of each dataset (DS#1, DS#2 and DS#3) for each model.

3.1.1 Gradient Boosted Tree

The Gradient Boosted Tree Algorithm applied to the DS#3 presented a mean absolute error of 1,141 and coefficient of determination of 0,19. The dynamic response is shown in Figure 5:

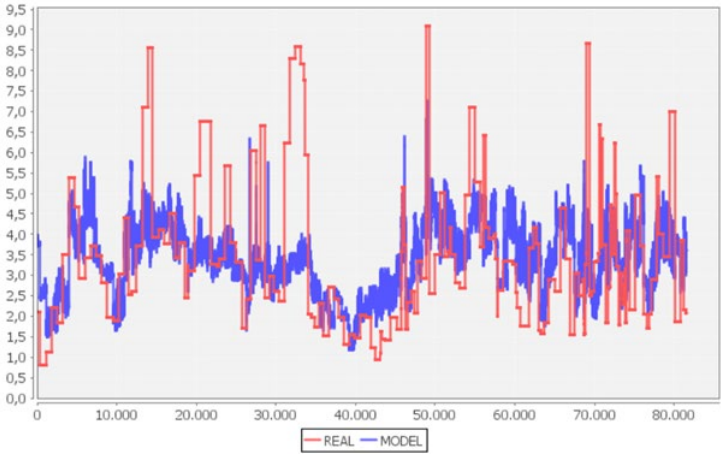

Figure 5 - Dynamic response of Gradient Boosted Tree algorithm
Source: Author's own figure

3.1.2 Random Forest

The Random Forest Tree Algorithm applied to the DS#3 presented a mean absolute error of 1,145 and coefficient of determination of 0,273. The dynamic response is shown in Figure 6:

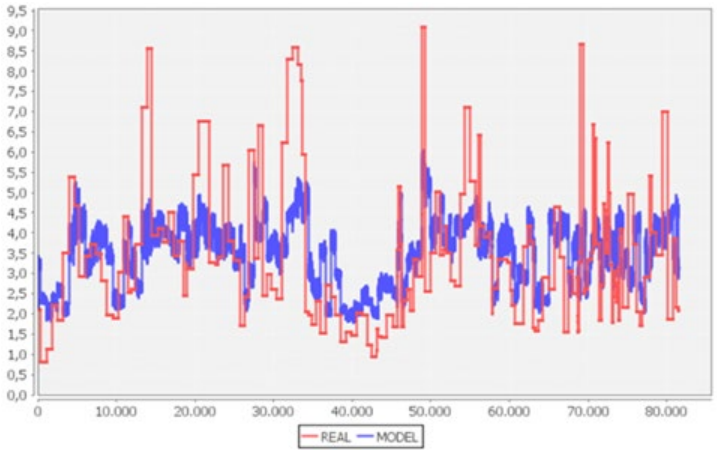

Figure 6 - Dynamic response of Random Forest Tree algorithm
Source: Author's own figure

3.1.3 Root Cause Analysis

The DS#2 had a better performance, and, using it, it was evaluated an accuracy of 94,346% in the test set, and the following rules to “silica meet the requirement”, described in Table2:

Table2 – Root Cause Rules

Rules
Output Stream - Fe Concentration > 66,805%
Feeding Flow Rate <= 2607,182 m^3/h
Conditioning Tank - Amina Flow <= 8,391 m^3/h

Therefore, 78,4% of the data with the characteristics listed in Table 2 corresponds to “silica meet the requirement”.

However, this analysis should be interpreted carefully. Firstly, because of one of the rules: the amine flow. According to this rule, if the amine rate is lower than 8,931m³/h, there is a high chance of the silica meet the specification (low silica percentage). But generally, the greater the amine flow the lower the silica percentage. So, this specific rule captured by this model could not be a cause, but a consequence: once the plant operator sees the high percentage of silica, he tries to increase the amine flow hoping that the silica percentage falls. And secondly, this analysis should be more explored using different residence time because the value used (one-hour lag) could not have been precise enough.

3.1.4 Multi-Layer Perceptron

The MLP applied to the DS#1 presented a mean absolute error of 0,7974 and R Squared of 0,4966. The expected output and the estimated value (or its moving average, just to show a cleaner dashboard) are depicted in Figure 7.

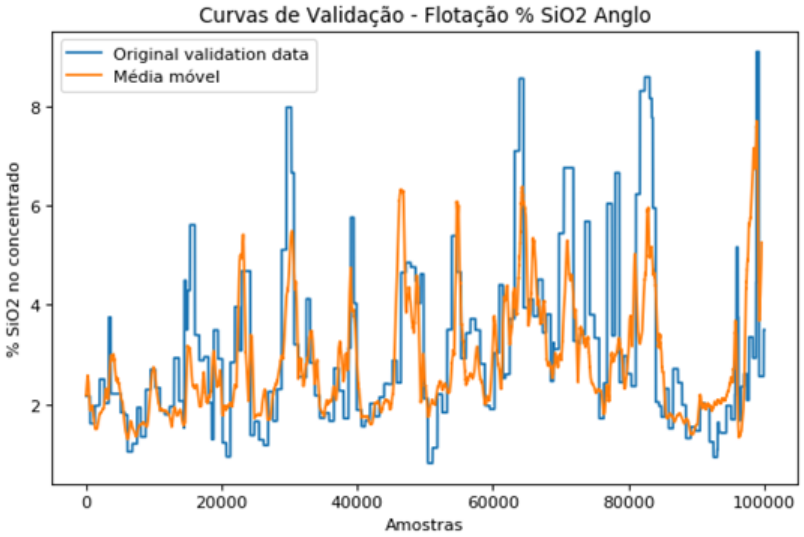

Figure 7 - Dynamic response of MLP algorithm
Source: Author’s own figure

3.2 Deployment

Mean Absolute Error was chosen as the suitable metric to define the model to be deployed mainly because, as it is absolute, it was easy to compare it to the target variable and to decide whether the model met the performance requirements. The values for each algorithm is presented in Table 3.

Table *3* - Mean Absolute Error for each model

Model	**Mean Absolute Error**
Gradient Boosted Tree	1,141
Random Forest	1,145
Multi-Layer Perceptron	0,7974

Based on the results achieved it was decided to deploy the MLP model. The application architecture built to estimate SiO_2 online is depicted in Figure 8. It has two main python scripts OPC Collector and Soft Sensor App. The first one is responsible for retrieving current values of process variables every 20 seconds from plant floor via OPC and store them in a relational database. The second one takes the last 05 minutes records, also including lab results, estimate silica percentage using weights and bias computed during the training phase and, finally, write the prediction to the PLC (the CPU that controls the industrial process).

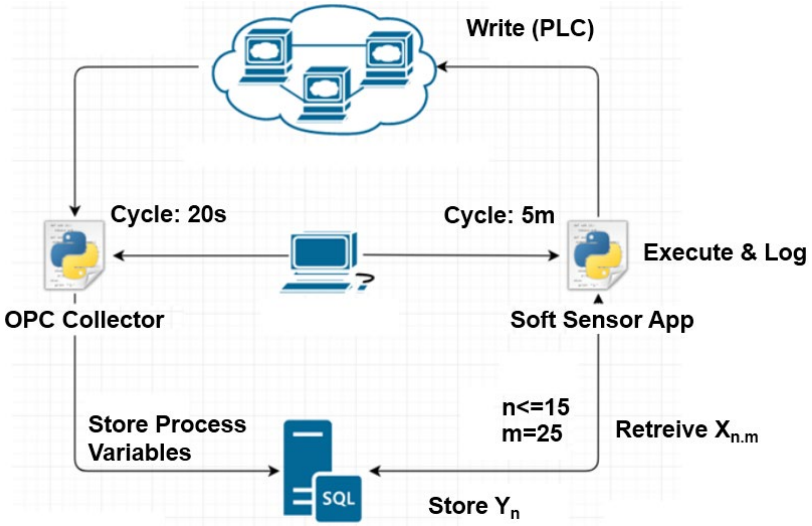

Figure 8 – Application Architecture
Source: Author's own figure

4 CONCLUSION

After application setup, the plant operator was able to take actions every 05 minutes relying on a prediction of SiO_2 with 0,8% of mean absolute error. Compared to the previous dead time, it was considered the great achievement of the project. And thus, it certified the power of this approach in industrial processes with high latency of laboratory results.

Some experiments showed that lab results related to the recirculation sampling significantly enhanced the performance of all the models. However, it was discarded because the sampling process is carried out manually and thus, it is not considered

reliable enough to take part as model input. Then, the secondary achievement of the study was to support the adoption of automated samplers on that stream.

As a future study, it was proposed to build and evaluate classification instead of regression models, by discretizing the value of SiO_2, which would widen the range of possibilities and could lead to better results.

REFERENCES

1 Kramer A., Gaulocher S., Martins M., Leal Filho L. (2012) Surface Tension Measurement for Optimization of Floating Control. Procedia Engineering. 46. 111–118. 10.1016/j.proeng.2012.09.453.

2 B. Shahbazi, S. ChehrehChelgani, S.S. Matin (2017) Prediction of froth flotation responses based on various conditioning parameters by Random Forest method. Colloids and Surfaces A 529 (2017) 936–941

3 Nakhaei F., Irannajad M. (2015) Application and comparison of RNN, RBFNN and MNLR approaches on prediction of flotation column performance. International Journal of Mining Science and Technology 25 (2015) 983–990

4 Despotović V., Trumić M. S., Trumić M. Ž (2017) Modeling and prediction of flotation performance using support vector regression. Recycling and Sustainable Development 10 (2017) 31-36

5 Mariscal G., Marbán Ó., Fernández C. (2010). A survey of data mining and knowledge discovery process models and methodologies. The Knowledge Engineering Review, 25(2), 137-166. doi:10.1017/S0269888910000032.

6 Piatetsky G. (2014) CRISP-DM, still the top methodology for analytics, data mining, or data science projects. Available at: https://www.kdnuggets.com/2014/10/crisp-dm-top-methodology-analytics-data-mining-data-science-projects.html. Accessed: 06/26/2018.

7 Pyle D. (1999). Data preparation for data mining. São Francisco: Morgan Kaufmann apud De Casto L. N., Ferrari D. G. (2016). Introdução à Mineração de Dados. Editora Saraiva.

8 Lasso (Statistics). Available at https://en.wikipedia.org/wiki/Lasso. Accessed: 06/26/2018

9 Dhiman V., Bedi R., Kumar M. (2017) Predictive Big Data Analysis for Machine Learning using Knime to solve many business challenges. Accessed: 06/26/2018. Available at: http://www.innovativeresearchpublication.com/singapore/paper%2045.pdf

10 Random Forest Learner – NodePit (2017) KNIME. Accessed: 06/06/2018. Available at: https://nodepit.com/node/org.knime.base.node.mine.treeensemble.node.randomforest.learner.classification.RandomForestClassificationLearnerNodeFactory