

**PROJECT REPORT**

MATH80629A - MACHINE LEARNING I: LARGE-SCALE DATA ANALYSIS AND DECISION MAKING

FEDERICO AGUILAR CASTELLANOS, 11296810

MICHEL CHAMOUN, 11306042

Due December 20th, 2021

# Abstract

Currently, companies are required to provide products that comply with rigorous industry standards, while at the same time meeting and exceeding their customers’ needs and expectations. Therefore, organizations often have strategies to improve the quality of their work processes and their products, since failure to address quality standards could result in customer dissatisfaction, a bad reputation, and problems with regulatory agencies (which is the case of more regulated industries, such as the mining industry). Additionally, proper quality control prevents the costs associated to inefficiencies, downtimes in production, and low revenues. As mentioned by Schmitt et al., the complexity of inspection and operation has increased; thus, requiring the application of the latest technology the era of Industry 4.0 has to offer.

During a mining operation the impurities (in the form of silica) present in iron ore concentrate are measured every hour. Early detection of such impurities could be beneficial for engineers as they would have more information on the process and the quality of their concentrate (i.e., final product). Additionally, reducing the number of impurities in concentrate could benefit the environment by reducing the amount of iron that goes in the waste stream of the process, denoted tailings. A ML model approach seems appropriate for the accurate prediction of the percent of silica in the iron ore concentrate. Two different models were considered for the predictive model, a multi-layered perceptron (MLP) neural network and a long short-term memory (LSTM) neural network. By considering a smaller lag timestep, the present project attempts to improve and evaluate the model developed by Pu et al. in their publication: Purities prediction in a manufacturing froth flotation plant: the deep learning techniques.

The results demonstrated that ML methods can predict the percentage of silica in the concentrate using a 20-second time-step with great accuracy. Additionally, the results also proved that time-series models performed better than their regression counterparts at predicting the % Silica Concentrate, the target variable.​

# Objectives

The main objective of the project was to develop a ML algorithm capable of predicting the percent of silica in the iron ore concentrate within a 20 second time frame. Due to the lack of trends in algorithms for manufacturing plants, especially mining plants, another objective is to clearly identify the strengths and weaknesses of the method (or combination of algorithms) that could potentially provide the most accurate prediction. Therefore, three different models of different complexities were considered to forecast the percent of silica in the iron ore concentrate. The results could have the potential to incentivise the application of ML techniques for manufacturing processes.

The data is publicly available; however, the number of known, unique contributors is limited to 58 – and most of them have focused on running either a regression analysis or a random forest generator. For this reason, it was interesting to evaluate the predictive power and performance of additional methods like neural networks, particularly an artificial recurrent neural network like Long Short-Term Memory (LSTM) and a multilayered perceptron classifier (MLP).

The main objectives of the project included,

* Generate a model capable of predicting the percent of silica present in a processed iron ore concentrate.
* Compare the different models' predictive accuracy.
* Evaluate the effect of measurement interpolation of sequential data on the model's performance.
* Provide valuable insight on the correlation and relationship between factors of the dataset.
* Provide recommendations for future research on the subject.

# Description of the Dataset

The data used for the project was retrieved from the Kaggle project proposal named 'Mining Quality Prediction'. There are a total of 737,453 sensor data entries measuring process variables such as air flow of the equipment, level of the liquid, feed flow of ore, among other properties. The original dataset was collected from machine sensors of the flotation columns of a mining operation, which were set to a sampling frequency of 20 seconds during a 6-month range spanning March 2017 to September 2017. The following is a brief description of the most relevant variables for the study,

Process input variables:

* **Date, hourly timestamp of the measurements**
* **% Iron Feed, percent of iron in the ore (raw material) being fed into the flotation process**
* **% Silica Feed, percent of silica in the ore (raw material) being fed into the flotation process**

Process output variables:

* **% Iron Concentrate, percent of iron measured at the end of flotation process**
* **% Silica Concentrate, percent of silica (impurity) measured at the end of flotation process**

## Datetime and Sampling Frequency

Of importance was the fact that the sensor readings corresponding to the variables Flotation Column Air Flow and Flotation Column Level were performed in 7 different flotation columns, numbered in the data set from 01 to 07. The sensor measurements were mostly recorded every 20 seconds; apart from those corresponding to the % Iron Feed, % Silica Feed, % Iron Concentrate, and % Silica Concentrate. This was of particular importance since the variable % Silica Concentrate was the variable of interest for the predictive model (i.e., the target variable).

The sensor timestamps of the original data did not reflect the 20 second sampling frequency, as the 180 measurements per hour were stored under the same hourly datetime label. Therefore, to prevent problems with duplicates, a new index reflecting the 20 second sampling frequency was created. It should be noted that only two instances appeared to have less than 180 measurements (1 hour / 20 s = 180 measurements),

* Timestamp 2017-03-10 01:00:00, which was missing 6 measurements per feature.
* Timestamp 2017-04-10 00:00:00, which was missing 1 measurement per feature.

The indexes corresponding to the missing measurements were removed from the new, generated 20 second indices. The methods of resampling and rescaling the data were adapted from those presented in the tutorial 'A programmer’s cleaning guide for messy sensor data' written by Xavier Ho and the work of Mark Koerner in 'Production Predictions from Sensor Data' (Ho, 2017).

## Feature Analysis

Only the variables % Iron Feed and % Silica Feed and % Iron Concentrate and % Silica Concentrate appeared to be sampled hourly - i.e., a mean frequency of 1, while the rest of the variables contain higher frequency measurements ranging from 153 to 178. From the Figure 1 below, the % Iron Feed, % Silica Feed, % Iron Concentrate, and % Silica Concentrate,

* There appears to be 4 main regions of interpolation for the variable % Iron Feed, and only 2 regions of interpolation for the variable % Iron Concentrate.
* The same regions of interpolation were observed for the % Silica Feed and % Silica Concentrate variables, which suggests that the interpolation of measurement was performed simultaneously, at the same hourly datetime stamp, for the Iron and Silica variables.

The plots suggest that both the % Iron Feed and % Silica Feed have linear interpolated values. Of importance, is the region between May 2017 and June 2017, where the plateau of the plot could be a strong indication of linear interpolation of measurements due to missing measurements. The latter observation is mainly based on the assumption that interpolated data will show little to no pixel noise in the plot.

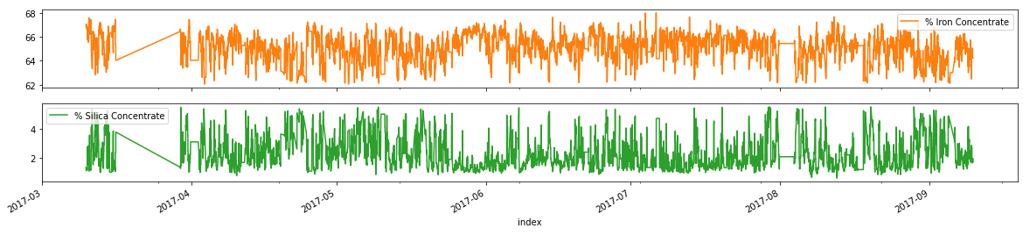
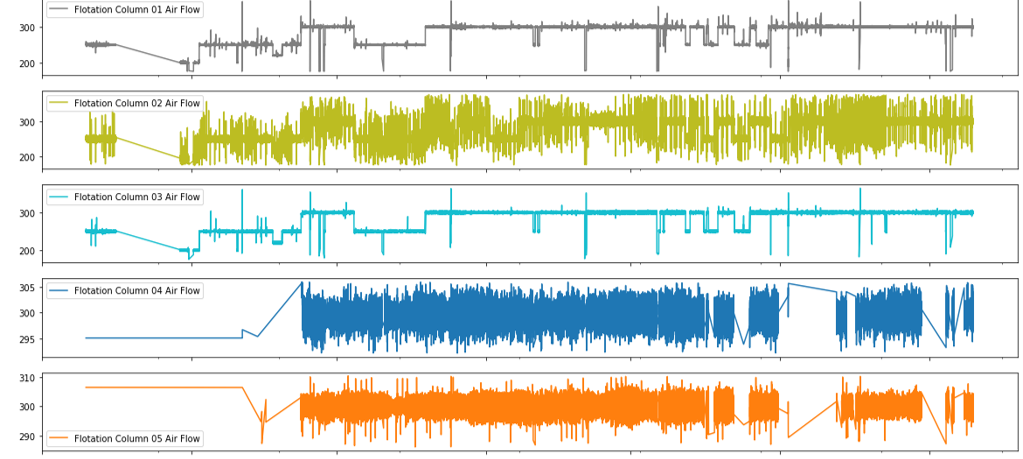
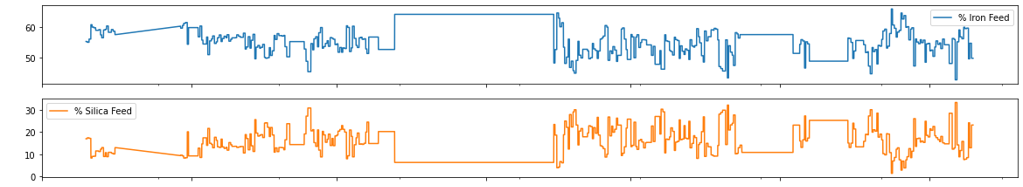


Figure : Sensor data of the features recorded between March 2017 and September 2017

By setting a threshold of two standard deviations for the z-score statistics, these values were removed from the data, and their corresponding values were linearly interpolated. Based on the previous observations and modifications, the period between April 2017 and May 2017, which showed the least amount of linearly interpolated measurements, appeared to be the most suitable candidate for the training and testing of the predictive models.

## Feature Correlation

There did not appear to be a strong correlation between variables other than,

* The % Silica Feed is strongly correlated to the % Iron Feed, which is to be expected as these materials are introduced simultaneously in the ore fed to the flotation process.
* The % Silica Concentrate is strongly correlated to the % Iron Concentrate, which is also to be expected as the main objective of the flotation process is the removal of impurities and extraction of the mineral. That is, the overall objective of the flotation process is the removal of silica and the extraction of iron.
* There is a correlation between the process variables of the different flotation columns. Specifically, the process variables, Air Flow and Level, of the Flotation Columns 01, 02, and 03 appear to be correlated; the same correlation appears to be present for the process variables of the Flotation Columns 04, 05, 06, 07.

Removing collinear data points would prevent model from overfitting; therefore, the variable % Iron Concentrate was removed since it was strongly correlated to the target variable % Silica Concentrate, and the variable % Silica Feed was dropped from the data as it was strongly correlated to the % Iron Feed.

Since there appeared to be a correlation between the process variables of the different flotation columns, only the process variables for the Flotation Column 02, Flotation Column 05, and Flotation Column 07 were kept for the training of the ML models. Figure 2 shows the correlation heatmap of the variables retained for fitting the ML models.

Chart, treemap chart

Description automatically generated

Figure : Feature correlation

# Methodology

The following sections provide a description of the ML models considered for the project including random forest regression models with and without time-series consideration, an MLP neural network, and an LSTM neural network. Additionally, the following sections also detail the results obtained from cross-validation and hyperparameter tunning.

## Random Forest

A random forest model was considered for the project since the predictions from the trees are averaged across all decision trees often results in a better performance than fitting any single tree or fitting a linear regression model.

The first fitted model was a random forest regressor that did not take into consideration the time dependency of the data. In other words, the data was considered as a normal data frame and not as time-series sensor measurements. It is worth mentioning that for this model the index of the data was still a date; however, the index did not contribute to the fitting and training of the model.

Regarding the hyperparameters of the model, the maximum depth of the individual trees was evaluated by doubling its value from 4 to 64 at each iteration. Then, the number of random features to consider at each split was varied. The tree possible cases considered were:

* 'auto': the number of random features is simply equal to the number of features of the model.
* 'sqrt': the number of random features is the square root of the number of features of the model.
* 'log2': the number of random features is the log base 2 of the number of features of the model.

The second model was a random forest regression with a time-dependency consideration, which required the transformation of the data from time-series to a supervised learning task. That is, to implement the second random forest model, the values of the input features were lagged (i.e., shifted) to predict the target variable. This function was further detailed in the methodology for the LSTM model as it was also used to fit the recurrent neural network. There is a temporal dependency between observations; thus, one must preserve that relation during cross-validation. For that reason, the time-series split method in the Sklearn Python library was used for the cross-validation of the Random Forest model.

Table 1 details the results obtained from the cross-validation of the model and the performance metric of the random forest models on the test set.

Table 1: Performance values for the random forest regression

|  |  |
| --- | --- |
|  | **MSE** |
| **Without time-series consideration** |  |
| Hyperparameter – optimal depth at 64 | 0.00349 |
| Hyperparameter – optimal number of features at auto | 0.00350 |
| Validation error | 0.00350 |
| Test error | 0.0470 |
| Accuracy of the model | 95.9128 % |
| **With time-series consideration** |  |
| Validation error | 0.04271 |
| Test error | 0.0398 |
| Accuracy of the model | 97.2918 %. |

Figure 3 illustrates the forecast vs. the actual value of % Silica Concentrate obtained from the random forest regression (a) without time-series consideration and (b) with time-dependency consideration.

Chart, line chart

Description automatically generated

Figure .a forecast vs actual value of % Silica concentrate using random forest - no time-series

A picture containing text, needle, lined, line

Description automatically generated

Figure 3.b forecast vs actual value of % Silica concentrate using random forest - time-series consideration

## Neural Networks

The other models considered were neural networks, an MLPRegressor and a Long Short-Term Memory (LSTM) model. Neural networks were considered due to their well-established reputation in handling time-series data and complex models.

Before training the model, the data was scaled using the MinMaxScaler from Sklearn Python library. This allows for the data to be in a certain range instead of varying on different scales. Scaling the data is critical to get better results and faster training process. It should be noted that the scaling of the features was also performed when fitting the random forest model.

The data was then split as following: 70% for training the neural network, 20% for validation (choosing the best structure) and 10% for testing. To build the neural network, an MLPRegressor from Sklearn was considered, and its performance was evaluated based on the mean squared error (MSE). These neural networks predicted the percentage of silica concentrate after the chemical process. As previously indicated, the variable % Iron Concentrate was not considered as a feature due to it is high correlation with the % Silica Concentrate, the target variable of the model.  Moreover, the % Iron Concentrate is a result of the froth flotation process and not an input; thus, adding this variable as a feature would not have given a lot of information.

### MLP

The data was treated as a data frame and not a time-series. The hyperparameters to be tuned are the number of hidden layers and the number of neurons in each hidden layer. We first started by finding the optimal number of neurons in each layer by training a 1-hidden layer neural network and varying the number of neurons from 2 till 128 by doubling it at each validation step. Afterwards, we tested the accuracy of networks having 2, 4, and 8 hidden layers. The best results were achieved with a neural network having 8 hidden layers and 128 neurons in each layer with an MSE of 0.1546 on the test set. The figure below illustrates the actual value and the forecast value of the %Silica Concentrate using MLP.

Chart, line chart

Description automatically generated

Figure real value vs forecast value of %Silica concentrate using MLP

### LSTM

The time-series data was transformed into samples of input of past observations and an output future observation, a common method used to convert a time series analysis into a supervised learning problem, as described by Jason Brownlee and Suggula Jagadeesh on their published articles (Brownlee, 2017).

Explicitly, the lagged observations (t-1) of the input variables were used to forecast the % Silica Concentrate at (t), where - 1 corresponds to a 20 second time lag. This approach is also presented in the documentation for the Tensorflow machine learning library, in which models make a set of predictions based on a window of consecutive samples from the data. It should be noted that the function below, series\_to\_supervised, is widely used for forecasting applications. The same function was used for the random forest model considering the time-series nature of the data.

Afterwards, the data was split into a train, test, and validation set. The splitting of the data was not random, since it had to be partitioned in a way to keep consecutive samples; that is, to maintain the time-dependency of the measurements in a time-series dataset. Therefore, the data was split using consecutive integer indices that would allow for a 60% training, 20% validation, and 20% testing split.

Initially, a univariate and a 2-variable LSTM model were considered to forecast the % Silica Concentrate. However, the performance of these two models was not adequate, generating a poor forecast and an elevated RMSE, which was the performance measurement of the model. For this reason, a multivariate LSTM model was considered as a suitable method to improve the forecast obtained by the models previously described in the report.

# Relationship to Previous Works

The interest of ML methods for quality prediction in mining operations has gained popularity in recent years; therefore, several papers have investigated ML quality-based predictions in a mining environment. In relation to the project, the following papers were of particular importance,

* The master's degree thesis of Eric Kwame Osei titled 'Machine Learning-based Quality Prediction in the Froth Flotation Process of Mining' (Osei, 2017)
* The paper by Pu et al titled 'Purities prediction in a manufacturing froth flotation plant: the deep learning techniques'. (Pu *et al*, 2020)

The first publication by Osei utilized the same dataset to generate solely regression models without considering the time-dependency of the features; explicitly, the models considered in the thesis project were a multiple linear regression, a random forest model, and an artificial neural network. For their thesis project, the datetime variable was eliminated from the analysis as non-predictive feature. Since their time-series consideration was different, this publication mainly served as a comparison metric to the feature analysis and variable importance for the preprocessing of the data. Explicitly, Osei detected a similar number of outliers during the exploration of the data; however, the author decided to remove these instances and retain only 521,188 observations, which was a different approach from the one implemented in the present project. In terms of feature correlation, the variable correlations previously discussed were also observed by the author in their published project; thus, serving as a validation to the preprocessing of the data. Additionally, the thesis also served to compare the optimal hyperparameter of the proposed regression models. In detail,

* The tunning of the hyperparameters of the random forest model revealed that the optimal number of selected predictors was 'auto', which corresponded to the number of features in the data. A similar result was obtained by Osei, since their cross-validation revealed that the optimal value for the selected predictors was approx. equal to the number of features.
* Cross-validation of the MLP model revealed that 128 neurons were optimal for the proposed network. Osei obtained a very similar result in their publication, in which their model started to exhibit diminishing returns, in terms of MSE, when the number of neurons exceeded 200.
* The RMSE values obtained for the proposed random forest and MLP models were lower than those presented in the thesis project. Explicitly, they obtained RMSE values of 0.5538 and 0.9081 for their neural network and random forest models, respectively. Meanwhile, the proposed MLP and random forest models recorded RMSE values of 0.3931 and 0.2167, respectively. This difference in performance metrics could be attributed to the data lost during their feature analysis, particularly by the elimination of observations and their decision to use measurements corresponding to a 12-minute sampling frequency vs. a 20-second sampling frequency.

The second publication by Pu et al, served as a comparison metric for the performance of the LSTM time-series model. In their study, they empirically chose a sequence length of 6 hours, and the architecture of their LSTM model consisted of three layers with 30 memory cells in each layer. However, the authors of the paper focused on the predictive task; thus, deciding to not consider hyperparameter tunning of their model. Other notable differences is that they decided to use a batch size of 32 and 60 training epochs, these values were different from the optimal values obtained during the cross-validation of the proposed LSTM model. Additionally, they decided to retain all the features of the original dataset. Overall, it was difficult to compare the performance of the proposed LSTM structure to the one presented by Pu et al in their publication due to their use of a different performance metric, the R-square value. Nevertheless, the authors reached a similar conclusion indicating that LSTM is an appropriate model for the quality-prediction of the froth flotation process.

Because the data is publicly available, other methods and authors explored the predictive capabilities of different ML models. However, most project submissions on the matter mainly focused on fitting a linear regression model without considering the time-series dependency of the data, which could reinforce our finding that a simpler model could accurately predict the % Silica Concentrate for the quality-prediction of the froth flotation process.

# Model Performance and Discussion

The base model consists of taking the average of the training set which results an MSE of 1.0621 for non-time series models and an MSE of 1.369 for time-series models. Both non-time series models performed better than the base models. In fact, the random forest model resulted an MSE of 0.047 on the test set and the MLP model had an MSE of 0.1546 on the same test set. Similarly, the two time-series models beat the base model with an MSE of 0.0398 for the random forest model and an MSE of 0.108 for the LSTM model.

In both time-series and non-time series contexts, random forest was more accurate than the neural network models. However, the LSTM model could be further improved by tuning it’s hyperparameters and performing cross-validation which requires immense computing power.

# Conclusions

The main findings and conclusions from the project include:

Time-series models performed better than the regression models at predicting the % Silica Concentrate.​

Random forest with time-series consideration performed slightly better than the LSTM neural network, and the random forest regression performed better than the MLP neural network.​

The better performance of both random forest models could be attributed to the susceptibility of neural networks to hyperparameter tunning, which was evidenced during the cross-validation of these models.​

Most importantly, all the considered models converged; thus, the objective to predict the percentage of silica in the concentrate using a 20-second time step was met with great accuracy.

# References

Brownlee J. How to Convert a Time Series to a Supervised Learning Problem in Python (2017). Machine Learning Mastery. URL: <https://machinelearningmastery.com/convert-time-series-supervised-learning-problem-python/>

Ho, X. A programmer’s cleaning guide for messy sensor data (2017) URL: <https://opensource.com/article/17/9/messy-sensor-data>

Osei, E. Machine Learning-based Quality Prediction in the Froth Flotation Process of Mining; Master’s Degree Thesis in Microdata Analysis. Dalarna University (2019). URL: <https://www.diva-portal.org/smash/get/diva2:1386720/FULLTEXT01.pdf>

Pu, Y., Szmigiel, A. & Apel, D.B. Purities prediction in a manufacturing froth flotation plant: the deep learning techniques. Neural Comput & Applic 32, 13639–13649 (2020). <https://doi.org/10.1007/s00521-020-04773-2>