# A Machine Learning approach to predict Length of Stay of vehicles in an inbound logistics operation

Victor Hugo Soares Pereira
*Graduate Program in Computational Modelling*
*Federal University of Juiz de Fora*
Juiz de Fora, Brazil
victor.pereira@engenharia.ufjf.br

Kaike Sa Teles Rocha Alves
*Graduate Program in Computational Modelling*
*Federal University of Juiz de Fora*
Juiz de Fora, Brazil
kaike.alves@engenharia.ufjf.br

Eduardo Pestana de Aguiar
*Dept. of Industrial and Mechanical Engineering*
*Federal University of Juiz de Fora*
Juiz de Fora, Brazil
eduardo.aguiar@engenharia.ufjf.br

***Abstract*—A large steel plant receives up to millions of tons of scrap metal through road transportation each year in its inbound logistics process. The Length of Stay (LOS) of vehicles is one of the most critical metrics that represent the performance of the unloading operation of raw materials. Accurately predicting this metric enables managers to make data-driven decisions at operational, tactical, and strategic levels. This study proposes implementing a Machine Learning (ML) approach for predicting the LOS of vehicles loaded with scrap metal in the inbound operation of a large Brazilian steel plant. The performance of five ML models - Linear Model Ridge, k-nearest neighbors Regressor, Gradient Boosting Regressor, Decision Tree Regressor, and ePL-KRLS-DISCO - was evaluated in terms of Root Mean Squared Error (RMSE), Mean Average Error (MAE), and execution time. The results are compared with the current method of prediction and statically validated through an Analysis of Variance (ANOVA) test. The ML approach applied in this study achieved better accuracy, reducing by 64% the RMSE, and has the potential to enable more reliable data-driven decisions for the company.**

***Index Terms*—Machine Learning, Steel Industry, Logistics, Regression models**

## I. Introduction

Steel production relies heavily on raw materials, mainly scrap metal [1]. A complex system moves millions of tons of scrap metal annually via road transportation to transport this material from scrapyards to steel plants. Accurately predicting the unloading times of these materials enables managers to make better-informed operational, tactical, and strategic decisions [2]. Such predictions can benefit the operation's performance by minimizing the negative impact of uncertainty in a context where information is carried throughout the supply chain [3].

Predicting the LOS metric for inbound operations is challenging because of the incredible variety of vehicles and material combinations to be considered, and external factors such as machinery breakdown, process interruptions, shift change, and first-time inside the plant vehicle drivers often create fluctuations that are difficult to predict. The current method used by a large steel plant in Brazil is to divide the net weight of each cargo by a flow rate factor plus a fixed amount of time. This method presents high values and variability of error metrics.

To address this issue, a novel Machine Learning (ML) approach is applied to predict the LOS metric using classic models from the literature. The chosen models, namely Linear Model Ridge, KNN Regressor, Gradient Boosting Regressor, and Decision Tree Regressor, have been selected from the Scikit-Learn library [4]. Also, to explore more complex and advanced models, ePL-KRLS-DISCO [5] was selected. Our approach is not limited to the weight of the cargo, like the current method. We propose a comprehensive view that considers different types of vehicles and materials, Key Process Indicators (KPIs), and time-related variables to improve the accuracy of the prediction. The models' results are compared to the current method's performance based on error metrics and execution time. The statistical relevance of this comparison is evaluated using an ANOVA test.

This paper presents the LOS metric and its importance to the company in Section II. It also discusses the current prediction method and presents the dataset. Section III briefly recalls the ML models implemented to predict the LOS metric. Section IV presents the tools used in this study and compares the error metrics of the ML models. Additionally, a statistical test evaluates the performance of the proposal. Finally, Section V presents the main conclusions and suggests future research.

## II. Problem formulation

### A. The dataset

The data for the inbound scrap metal process was collected by extracting several months' worth of information from the company's database. In total, 23,974 observations were recorded and divided into nine datasets, one for each month. Each observation represents a truck whose cargo was unloaded at the steel plant.

TABLE I
EXAMPLE OF THE DATASET

| Attributes | Vehicle 1 | Vehicle 2 | Vehicle 3 |
|---|---|---|---|
| **day_of_the_week** | 2 | 6 | 5 |
| **No._half_of_the_month** | 1 | 1 | 1 |
| **group_of_vehicle** | 1 | 0 | 0 |
| **type_of_vehicle** | 75 | 127 | 127 |
| **bin_sweeping** | 1 | 1 | 1 |
| **sweeping_time** | 2.58 | 0.88 | 0.76 |
| **unload_location** | 27 | 14 | 14 |
| **id_material** | 32 | 30 | 30 |
| **multi_material** | 1 | 1 | 1 |
| **net_weight** | 48,220 | 72,960 | 72,920 |
| **amount_of_vehicles_day** | 119 | 95 | 125 |
| **amount_of_vehicles_inside** | 40 | 40 | 40 |
| **LOS** | 6.06 | 5.68 | 5.37 |

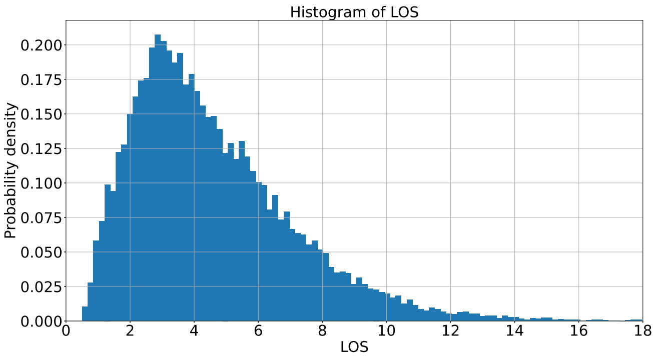

Fig. 1. Histogram of LOS values

Table I presents a random sample of three vehicles to describe the dataset used in this study. The first column shows 12 attributes that characterize the cargo, the vehicle, and the current state of the inbound process. The following columns are example values for each of these attributes of the vehicles. Each observation collected from the company's database contains the twelve attributes and the LOS time.

Table II presents mean values, standard deviation (SD), and the 25th, 50th, and 75th percentiles of the LOS metric, divided into nine datasets. The LOS metric is recorded in hours.

TABLE II
DESCRIPTIVE STATISTICS OF THE LOS METRIC

| Dataset | Size | Mean | SD | 25th | 50th | 75th |
|---|---|---|---|---|---|---|
| DS_1 | 3,009 | 4.93 | 6.32 | 2.92 | 4.30 | 6.05 |
| DS_2 | 3,026 | 4.74 | 3.04 | 2.86 | 4.18 | 5.94 |
| DS_3 | 3,577 | 4.91 | 2.51 | 3.09 | 4.45 | 6.23 |
| DS_4 | 2,889 | 4.94 | 3.68 | 3.01 | 4.32 | 6.36 |
| DS_5 | 2,492 | 4.56 | 5.49 | 2.75 | 3.98 | 5.72 |
| DS_6 | 2,884 | 4.48 | 6.59 | 2.40 | 3.69 | 5.76 |
| DS_7 | 1,814 | 4.67 | 2.91 | 2.54 | 3.89 | 6.24 |
| DS_8 | 2,196 | 3.85 | 2.52 | 2.15 | 3.27 | 5.03 |
| DS_9 | 2,087 | 3.81 | 2.64 | 2.45 | 3.36 | 4.74 |
| Entire Dataset | 23,974 | 4.60 | 4.37 | 2.71 | 3.99 | 5.84 |

### *B. The LOS metric*

The Length of Stay metric is a KPI constantly monitored by the managers of the steel plant. It represents the total time spent by a vehicle loaded with raw material to unload its cargo. The importance of this indicator relies on the fact that it directly impacts productivity, safety, and laws that protect truck drivers against long hours of unloading cargo.

The histogram of the LOS metric is presented in Figure 1. Based on the histogram, it's possible to observe a wide range of values for LOS, from less than an hour up to 19 hours.

The descriptive statistics of the LOS metric in Table II show high standard deviation values when compared to the mean values for each dataset. The main causes for variation in the total time a vehicle spends at the plant can be attributed to several factors. These factors involve a range of elements, including the variety of vehicles and materials combinations to be considered, machinery breakdown, process interruptions and overload, shift change, queues, and first-time inside the plant vehicle drivers. The high variability associated with this metric brings uncertainty to the daily operation, such as planning resources like machinery and workers. Also, an example of the impact of this uncertainty on a tactical level is related to the scrap metal's purchase planning process, which is directly influenced by the capacity of the plant to receive scrap metal.

### *C. Current method of prediction*

Equation 1 represents the current estimation method of the LOS metric. The method consists of the division of the net weight (ton) of each cargo ($W$) by a flow rate factor (ton/minute) ($F$) plus a fixed amount of time ($T$) to absorb other activities of the process, such as clearance, weighing and motion inside the plant. The flow rate factor is typically established as an average value, not subject to frequent reassessment, and is applied uniformly across all vehicles.

$$LOS = \frac{1}{60}\left(\frac{W}{\mathrm{F}} + \mathrm{T}\right) \qquad (1)$$

Table III summarizes two error measures of the current method of LOS prediction: Root Mean Square Error (RMSE) and Mean Absolute Error (MAE).

TABLE III
ERROR METRICS FOR THE CURRENT METHOD OF LOS PREDICTION

| Dataset | RMSE | MAE |
|---|---|---|
| DS_1 | 6.37 | 2.05 |
| DS_2 | 3.12 | 1.95 |
| DS_3 | 2.55 | 1.88 |
| DS_4 | 3.74 | 2.00 |
| DS_5 | 5.53 | 1.94 |
| DS_6 | 6.63 | 2.25 |
| DS_7 | 2.95 | 2.24 |
| DS_8 | 2.65 | 1.97 |
| DS_9 | 2.70 | 1.65 |
| Mean | 4.03 | 1.99 |

The RMSE of the method presents high variability among the datasets, with values ranging from 2.55 hours up to 6.63 hours. The MAE presents more stability than the RMSE but is notably smaller. This discrepancy in the magnitude of the error metrics shows that the current predictions have large residues due to the quadratic nature of the RMSE. [6].

## III. Proposed models

### A. Linear Ridge Regression

Linear Regression is a widely used statistical method that aims to predict the value of a target variable based on a linear combination of the input features. The method, proposed by [7], is based on the Ordinary Least Squares (OLS) approach, which minimizes the residual sum of squares between the observed targets in the dataset and the targets predicted by the linear approximation. However, the OLS method can be prone to imprecision when the input features are highly correlated, a situation known as collinearity.

In summary, Linear Ridge Regression is a useful method for addressing the problem of collinearity in Linear Regression, by applying a penalty on the size of the coefficients to make them more robust. It is an important technique for data analysts and statisticians to be familiar with, especially when working with datasets that have highly correlated features.

### B. k-Nearest Neighbors Regressor

The k-Nearest Neighbors algorithm is a non-parametric method, meaning it does not make any assumptions about the underlying distribution of the data. This makes it a versatile algorithm that can be applied to various regression problems [8], [9], [10]. One of the key advantages of the kNN algorithm is its ability to handle both continuous and categorical variables, making it well-suited for problems with mixed data types.

The application of kNN is based on the idea that data generated by a specific process may exhibit recurring patterns of behavior [10]. The kNN algorithm, first proposed by [11], predicts a new value by comparing it to the k most similar past patterns and using that information to make a prediction.

### C. Gradient Boosting Regressor

A boosting process is a method for improving the accuracy of learning algorithms by fitting an initial model to the data and then building a second model focused on accurately predicting the cases in which the first had a bad performance [12]. Proposed by [13], the Gradient Boosting Regressor uses a differentiable loss function (e.g. squared error) to guide an additive method of creating weak learners in a greedy way, following a gradient descent procedure and, thus, minimizing loss.

According to [14], the performance of the Gradient Boosting Regressor can be affected by three parameters: maximum number of trees, learning rate, and max depth of the tree. The best combination of the parameters enables the optimal result of the model. The first refers to the total number of trees (i.e. weak learners) integrated into Gradient Boosting Regressor. The second parameter sets the contribution of each weak learner to the final results, with values between 0 and 1. The third parameter expresses the complexity of the tree. Gradient Boosting Regressor is a strong learner formed by the combination of weak learners. Therefore, the max depth of each tree must be controlled in order to limit the complexity of the whole system [14].

### D. Decision Tree Regressor

Decision Trees are among the earliest statistical algorithms to be implemented in electronic form and are widely used for regression problems [15], [16], [17]. The primary characteristic of this model is its recursive subsetting of the data according to the values of the predictors, which progressively narrows the possible values into decision nodes until the model reaches a prediction (leaf nodes).

Also, decision trees are widely used in regression problems due to their simplicity and interpretability. The recursive subsetting of the data, the concepts of splitting and pruning, and the criteria used to split the data are some of the key aspects of decision trees. However, it is important to note that decision trees can be prone to overfitting, especially when the tree becomes too deep or detailed, so it is necessary to use techniques such as pruning to control the complexity of the model.

### E. ePL-KRLS-DISCO

Fuzzy logic is a mathematical approach based on the concept of degrees of truth, rather than the traditional binary boolean logic of true or false [18]. It is used to deal with the uncertainty and vagueness that is often present in real-world systems and problems.

Fuzzy rule-based systems, like the ePL-KRLS-DISCO model [5], express their knowledge base using a collection of fuzzy if-then rules. These rules are used to infer the output based on the input. Each rule has an antecedent, which defines the conditions for the rule to be activated, and a consequent, which defines the output when the rule is activated. The antecedent and consequent are defined using fuzzy sets, providing a flexible and intuitive way to express the knowledge of the system.

The ePL-KRLS-DISCO [5] model also incorporates Evolving Participatory Learning (ePL), which is an evolution of the Participatory Learning (PL) method proposed by Lima et al. [19]. PL is a recursive unsupervised clustering algorithm that implements convex combinations between input data and the closest cluster center. Additionally, the model uses Kernel Recursive Least Squares (KRLS), a nonlinear version of the recursive least squares algorithm, which performs linear regression in a high-dimensional feature space using kernel methods [20].

Moreover, the model incorporates Distance Correlation (DISCO). This method forms the rules with a reduced standard deviation, which improves the quality of the clusters and the models' capacity for learning. This is an addition to the algorithm proposed by Alves and Aguiar [5] and has substantially superior performance compared to previous models. The ePL-KRLS-DISCO model can perform precise simulations even with complex data, making it a powerful tool for data analysts and researchers.

## IV. Experimental results

The five ML models mentioned in Section III were trained and tested using the free version of the Google Colaboratory platform, a serverless Jupyter notebook environment for interactive development [21]. The Python Notebook file and the datasets used for this study can be found at: https://bit.ly/33qoJZe

TABLE IV
Results of the predictions

| DS | ML Model | RMSE | MAE | Time (s) |
|---|---|---|---|---|
| 1 | LR Regression | 1.49 ± 0.14 | 1.00 ± 0.06 | 0.02 ± 0.01 |
| | KNN Regressor | 3.01 ± 0.13 | 1.41 ± 0.02 | 0.02 ± 0.00 |
| | GBR | 1.36 ± 0.12 | 0.92 ± 0.04 | 1.78 ± 0.33 |
| | DT Regressor | 1.54 ± 0.11 | 1.01 ± 0.04 | 0.03 ± 0.02 |
| | **ePL-KRLS-DISCO** | **1.31 ± 0.11** | **0.97 ± 0.07** | **651.99 ± 750.32** |
| 2 | LR Regression | 1.59 ± 0.11 | 1.03 ± 0.04 | 0.01 ± 0.01 |
| | KNN Regressor | 2.88 ± 0.18 | 1.35 ± 0.04 | 0.02 ± 0.00 |
| | GBR | 1.52 ± 0.11 | 0.98 ± 0.04 | 1.85 ± 0.48 |
| | DT Regressor | 1.64 ± 0.10 | 1.06 ± 0.04 | 0.01 ± 0.00 |
| | **ePL-KRLS-DISCO** | **1.49 ± 0.18** | **1.07 ± 0.05** | **558.88 ± 682.01** |
| 3 | LR Regression | 1.46 ± 0.10 | 0.97 ± 0.04 | 0.01 ± 0.00 |
| | KNN Regressor | 2.87 ± 0.17 | 1.36 ± 0.04 | 0.01 ± 0.00 |
| | GBR | 1.45 ± 0.09 | 0.96 ± 0.03 | 2.22 ± 0.67 |
| | DT Regressor | 1.49 ± 0.09 | 1.00 ± 0.04 | 0.02 ± 0.00 |
| | **ePL-KRLS-DISCO** | **1.40 ± 0.15** | **1.04 ± 0.10** | **600.13 ± 771.50** |
| 4 | LR Regression | 1.69 ± 0.10 | 1.06 ± 0.04 | 0.02 ± 0.01 |
| | KNN Regressor | 3.10 ± 0.15 | 1.42 ± 0.04 | 0.02 ± 0.01 |
| | GBR | 1.75 ± 0.12 | 1.05 ± 0.04 | 1.96 ± 0.68 |
| | DT Regressor | 1.71 ± 0.10 | 1.07 ± 0.04 | 0.01 ± 0.00 |
| | **ePL-KRLS-DISCO** | **1.55 ± 0.19** | **1.11 ± 0.07** | **443.58 ± 568.89** |
| 5 | LR Regression | 1.55 ± 0.10 | 1.00 ± 0.04 | 0.02 ± 0.01 |
| | KNN Regressor | 2.91 ± 0.21 | 1.36 ± 0.05 | 0.02 ± 0.01 |
| | GBR | 1.47 ± 0.11 | 0.95 ± 0.04 | 1.56 ± 0.46 |
| | DT Regressor | 1.57 ± 0.11 | 1.00 ± 0.04 | 0.01 ± 0.00 |
| | **ePL-KRLS-DISCO** | **1.44 ± 0.16** | **1.03 ± 0.05** | **375.96 ± 474.09** |
| 6 | LR Regression | 1.71 ± 0.11 | 1.06 ± 0.03 | 0.01 ± 0.00 |
| | KNN Regressor | 2.53 ± 0.17 | 1.23 ± 0.04 | 0.01 ± 0.00 |
| | **GBR** | **1.24 ± 0.11** | **0.80 ± 0.03** | **1.71 ± 0.45** |
| | DT Regressor | 1.63 ± 0.08 | 1.04 ± 0.03 | 0.01 ± 0.00 |
| | ePL-KRLS-DISCO | 1.48 ± 0.16 | 1.06 ± 0.05 | 420.83 ± 544.15 |
| 7 | LR Regression | 1.82 ± 0.15 | 1.09 ± 0.05 | 0.04 ± 0.02 |
| | KNN Regressor | 3.49 ± 0.31 | 1.49 ± 0.07 | 0.03 ± 0.01 |
| | GBR | 1.76 ± 0.18 | 1.03 ± 0.06 | 1.16 ± 0.35 |
| | DT Regressor | 1.82 ± 0.14 | 1.10 ± 0.05 | 0.01 ± 0.00 |
| | **ePL-KRLS-DISCO** | **1.60 ± 0.25** | **1.11 ± 0.06** | **171.51 ± 221.97** |
| 8 | LR Regression | 1.55 ± 0.11 | 0.99 ± 0.04 | 0.01 ± 0.00 |
| | KNN Regressor | 2.93 ± 0.25 | 1.34 ± 0.06 | 0.01 ± 0.00 |
| | GBR | 1.54 ± 0.10 | 0.96 ± 0.04 | 1.61 ± 0.52 |
| | DT Regressor | 1.48 ± 0.10 | 0.97 ± 0.04 | 0.01 ± 0.00 |
| | **ePL-KRLS-DISCO** | **1.38 ± 0.13** | **1.00 ± 0.05** | **266.15 ± 344.02** |
| 9 | LR Regression | 1.27 ± 0.11 | 0.90 ± 0.04 | 0.02 ± 0.01 |
| | KNN Regressor | 2.42 ± 0.21 | 1.24 ± 0.06 | 0.02 ± 0.01 |
| | GBR | 1.34 ± 0.14 | 0.90 ± 0.05 | 1.45 ± 0.41 |
| | DT Regressor | 1.34 ± 0.12 | 0.91 ± 0.04 | 0.01 ± 0.00 |
| | **ePL-KRLS-DISCO** | **1.27 ± 0.11** | **0.94 ± 0.05** | **287.04 ± 366.55** |

Since most transports' characteristics were categorical variables (Table I), it was necessary to perform some preprocessing in the database to have only integers and floats as inputs to the models. For this, the technique Label-Encoding [22] was implemented, converting the categorical variables into an associated integer number. To preserve the company's sensitive information, the original data is not shown in this study. Due to the quality of the data extracted, with a 0% missing rate, no additional work was needed to replace missing values.

Then, each of the 9 datasets was separated into random training and test subsets on the ratio of 85:15 using the function train_test_split from the scikit-learn Python ML library [4]. A parameter of this function, named random_state, is a pseudo-random number generator and controls the shuffling applied to the data before applying the split [4]. The algorithms were put inside a loop structure, altering the random_state parameter from 1 to 50. The results of each iteration were recorded to compare the average outcome of each model and its standard deviation.

TABLE V
Means comparison

| DS | # Rank | ML Model | Differs from |
|---|---|---|---|
| 1 | 1 | ePL-KRLS-DISCO | 2, 3, 4, 5 |
| | 2 | Gradient Boosting Regressor | 1, 3, 4, 5 |
| | 3 | Linear Model Ridge | 1, 2, 4, 5 |
| | 4 | Decision Tree Regressor | 1, 2, 3 ,5 |
| | 5 | KNN Regressor | 1, 2, 3, 4 |
| 2 | 1 | ePL-KRLS-DISCO | 2, 3, 4, 5 |
| | 2 | Gradient Boosting Regressor | 1, 3, 4, 5 |
| | 3 | Linear Model Ridge | 1, 2, 5 |
| | 4 | Decision Tree Regressor | 1, 2, 5 |
| | 5 | KNN Regressor | 1, 2, 3, 4 |
| 3 | 1 | ePL-KRLS-DISCO | 2, 3, 4, 5 |
| | 2 | Gradient Boosting Regressor | 1, 5 |
| | 3 | Linear Model Ridge | 1, 5 |
| | 4 | Decision Tree Regressor | 1, 5 |
| | 5 | KNN Regressor | 1, 2, 3, 4 |
| 4 | 1 | ePL-KRLS-DISCO | 2, 3, 4, 5 |
| | 2 | Linear Model Ridge | 1, 5 |
| | 3 | Decision Tree Regressor | 1, 5 |
| | 4 | Gradient Boosting Regressor | 1, 5 |
| | 5 | KNN Regressor | 1, 2, 3, 4 |
| 5 | 1 | ePL-KRLS-DISCO | 2, 3, 4, 5 |
| | 2 | Gradient Boosting Regressor | 1, 3, 4, 5 |
| | 3 | Linear Model Ridge | 1, 2, 5 |
| | 4 | Decision Tree Regressor | 1, 2, 5 |
| | 5 | KNN Regressor | 1, 2, 3, 4 |
| 6 | 1 | Gradient Boosting Regressor | 2, 3, 4, 5 |
| | 2 | ePL-KRLS-DISCO | 1, 3, 4, 5 |
| | 3 | Decision Tree Regressor | 1, 2, 4 ,5 |
| | 4 | Linear Model Ridge | 1, 2, 3, 5 |
| | 5 | KNN Regressor | 1, 2, 3, 4 |
| 7 | 1 | ePL-KRLS-DISCO | 2, 3, 4, 5 |
| | 2 | Gradient Boosting Regressor | 1, 5 |
| | 3 | Decision Tree Regressor | 1, 5 |
| | 4 | Linear Model Ridge | 1, 5 |
| | 5 | KNN Regressor | 1, 2, 3, 4 |
| 8 | 1 | ePL-KRLS-DISCO | 2, 3, 4, 5 |
| | 2 | Decision Tree Regressor | 1, 3, 4, 5 |
| | 3 | Gradient Boosting Regressor | 1, 2, 5 |
| | 4 | Linear Model Ridge | 1, 2, 5 |
| | 5 | KNN Regressor | 1, 2, 3, 4 |
| 9 | 1 | ePL-KRLS-DISCO | 3, 4, 5 |
| | 2 | Linear Model Ridge | 3, 4, 5 |
| | 3 | Decision Tree Regressor | 1, 2, 5 |
| | 4 | Gradient Boosting Regressor | 1, 2, 5 |
| | 5 | KNN Regressor | 1, 2, 3, 4 |

### A. Evaluation method

The evaluation of the models was measured with two error measures - Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). Also another relevant metric to compare the performance of the models is computational complexity. The model execution time is usually a good representation since faster computational speed increases the possibility of algorithm deployment. [23].

### B. Models' results

Table IV summarizes the results of the models for each dataset, evaluated with the metrics presented above, and Figure

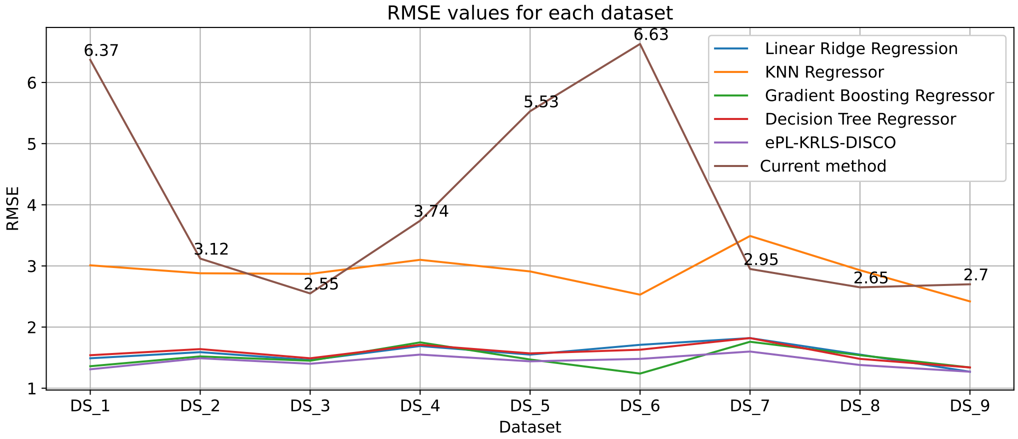

Fig. 2. Plot of the RMSE values of the models and current method of prediction

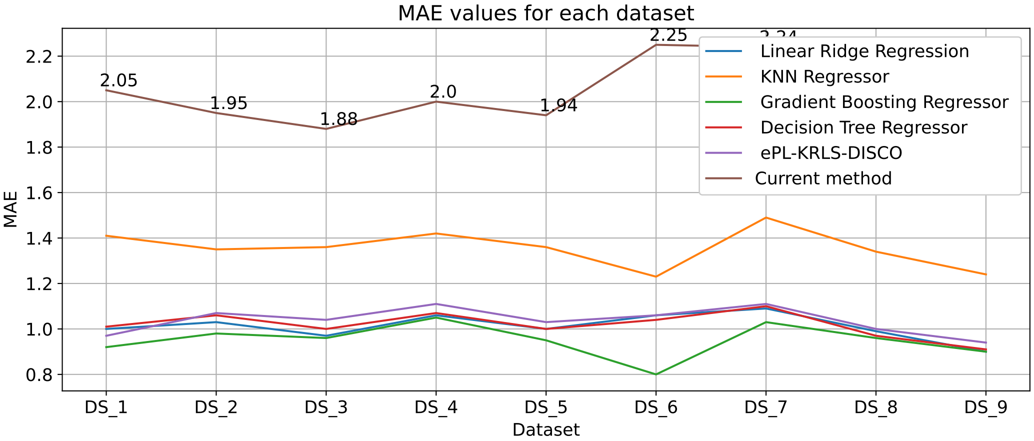

Fig. 3. Plot of the MAE values of the models and current method of prediction

2 shows the graphics of the predictions.

To statistically validate the results, a One-Way ANOVA test was performed. According to [24], the procedure uses the variances of the groups to determine whether the means are different. The comparison of variance between group means against the variance within groups works to determine whether the groups are all part of a larger population or distinct populations with different characteristics. The null hypothesis states that all populations' means are equal, while the alternative hypothesis states that at least one is different [25]. The categorical factor used in the test is the ML models, while the continuous response variable is the RMSE of each model for each dataset.

Considering a significance level ($\alpha$) of 0.05. If the $p-value$ is lower than $\alpha$, there's not enough information to conclude the null hypothesis is true, and the statement that all the models have equal accuracy is rejected.

For all datasets, *p-value* $< 0.001$ and thus, at least one of the models has a different accuracy. Analyzing the means comparison outcome from One-way ANOVA, it's possible to identify which models each model has different accuracy. Table V presents the ranking of the models in terms of better accuracy for each dataset, and Figure 2 shows the graphics of the predictions. Also, it presents for each model the models to which it does not overlap the confidence interval for its RMSE value.

The results of the statistical test show that ePL-KRLS-DISCO demonstrated the best accuracy for datasets 1, 2, 3, 4, 5, 7, 8, and 9, compared to the other presented ML Models, with a 95% confidence level, achieving the lowest values of RMSE and not overlapping the confidence interval of its mean RMSE with any other model. Considering all datasets, the model's mean value of RMSE presents a reduction of 64% when compared to the current prediction method. Regarding dataset 6, Gradient Boosting Regressor demonstrated the best result, with an RMSE of 1.24 ± 0.11, followed by ePL-KRLS-DISCO, with an RMSE of 1.48 ± 0.16. The KNN Regressor performed the worst results of accuracy in all datasets. Linear

Model Ridge and Decision Tree Regressor achieved average results compared to the other models. Also, according to the One-way ANOVA Test, these models' accuracy results are statistically equal for datasets 2, 3, 4, 5, and 7.

Regarding the computational cost, estimated by the algorithm's execution time, ePL-KRLS-DISCO presented values starting at 171.51 ± 221.97 seconds up to 651.99 ± 750.32 seconds. Gradient Boosting Regressor performed the second-worst results with values ranging from 1.16 ± 0.35 seconds to 2.22 ± 0.67 seconds. All other models presented execution time values lower than 0.04 seconds with almost zero standard deviation.

### C. Discussions

Despite the KNeighbors Regressor, the results of the ML models tested show that it can reduce the variance and achieve significantly lower RMSE values than the currently used method, which presents values ranging from 2.55 hours up to 6.37 hours. RMSE values considerably higher than MAE are a good indicator of significant residues in the prediction due to the quadratic nature of the RMSE [6]. The ML approach also presented lower values of MAE in all datasets tested, with errors inferior to 1 hour. The results of this approach reveal the potential to enable more reliable data-driven decisions regarding the inbound process of scrap metal.

## V. Conclusions

This study presented the results of a ML approach to predicting vehicles' Length of Stay for the inbound operation of scrap metal in a Brazilian steel plant. The results' simulations show a feasible solution to improve the accuracy of the LOS prediction issue.

The current prediction method is a simple model that only considers one attribute to make a prediction. Therefore, the ML approach is justifiable by the importance of the KPI to the company, the amount of data available that needs to be analyzed, and the need to have more reliable and accurate values for the LOS prediction.

The results of this study presented five ML models that achieved lower metrics of error than the current method used. This represents an opportunity for the company to consider using ML to predict important indicators and enable more robust data-driven decisions.

Future work includes evaluating other related data sources to improve accuracy (e.g., scrap metal purchase plan data, stock level, and weather conditions). Also, this study is an initial step for implementing a decision model based on the predicted LOS metric. In the State-Of-The-Art of 4.0 Industry, a decision model could optimize the process in real time. In the context of the inbound process of scrap metal, entrance anticipation, selection of unloading location, and priority pass of vehicles are routine decisions that could be optimized and automated.

## References

[1] O. V. Yuzov and A. M. Sedykh, *Metallurgist*, vol. 47, no. 5/6, p. 201–205, 2003.

[2] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*. OTexts, 2018.

[3] X. Zhao and J. Xie, "Forecasting errors and the value of information sharing in a supply chain," *International Journal of Production Research*, vol. 40, no. 2, pp. 311–335, 2002. [Online]. Available: https://doi.org/10.1080/00207540110079121

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[5] K. S. T. R. Alves and E. P. de Aguiar, "A novel rule-based evolving fuzzy system applied to the thermal modeling of power transformers," *Applied Soft Computing*, vol. 112, p. 107764, 2021.

[6] T. Chai and R. R. Draxler, "Root mean square error (rmse) or mean absolute error (mae)," *Geoscientific Model Development Discussions*, vol. 7, no. 1, pp. 1525–1534, 2014.

[7] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970. [Online]. Available: http://www.jstor.org/stable/1267351

[8] C. Hu, G. Jain, P. Zhang, C. Schmidt, P. Gomadam, and T. Gorka, "Data-driven method based on particle swarm optimization and k-nearest neighbor regression for estimating capacity of lithium-ion battery," *Applied Energy*, vol. 129, pp. 49–55, 2014.

[9] S. Kohli, G. T. Godwin, and S. Urolagin, "Sales prediction using linear and knn regression," in *Advances in Machine Learning and Computational Intelligence*. Springer, 2021, pp. 321–329.

[10] T. Ban, R. Zhang, S. Pang, A. Sarrafzadeh, and D. Inoue, "Referential knn regression for financial time series forecasting," in *International Conference on Neural Information Processing*. Springer, 2013, pp. 601–608.

[11] S. Yakowitz, "Nearest-neighbour methods for time series analysis," *Journal of Time Series Analysis*, vol. 8, no. 2, pp. 235–247, 1987.

[12] R. E. Schapire, "A brief introduction to boosting," in *Ijcai*, vol. 99. Citeseer, 1999, pp. 1401–1406.

[13] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of Statistics*, pp. 1189–1232, 2001. [Online]. Available: https://www.jstor.org/stable/2699986

[14] X. Zhan, S. Zhang, W. Y. Szeto, and X. Chen, "Multi-step-ahead traffic speed forecasting using multi-output gradient boosting regression tree," *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, vol. 24, pp. 125–141, 3 2020.

[15] J. M. Scavuzzo, F. Trucco, M. Espinosa, C. B. Tauro, M. Abril, C. M. Scavuzzo, and A. C. Frery, "Modeling dengue vector population using remotely sensed data and machine learning," *Acta Tropica*, vol. 185, pp. 167–175, 9 2018.

[16] H. Saghafi and M. Arabloo, "Modeling of co2 solubility in mea, dea, tea, and mdea aqueous solutions using adaboost-decision tree and artificial neural network," *International Journal of Greenhouse Gas Control*, vol. 58, pp. 256–265, 3 2017.

[17] S. Choudhury, D. N. Thatoi, J. Hota, and M. D. Rao, "Predicting crack through a well generalized and optimal tree-based regressor," *International Journal of Structural Integrity*, vol. 11, pp. 783–807, 9 2020.

[18] W. Pedrycz, *Fuzzy control and fuzzy systems*. Research Studies Press Ltd., 1993.

[19] E. Lima, M. Hell, R. Ballini, and F. Gomide, "Evolving fuzzy modeling using participatory learning," *Evolving intelligent systems: methodology and applications*, pp. 67–86, 2010.

[20] Y. Engel, S. Mannor, and R. Meir, "The kernel recursive least-squares algorithm," *IEEE Transactions on signal processing*, vol. 52, no. 8, pp. 2275–2285, 2004.

[21] E. Bisong, *Google Colaboratory*. Berkeley, CA: Apress, 2019, pp. 59–64. [Online]. Available: https://doi.org/10.1007/978-1-4842-4470-8_7

[22] J. T. Hancock and T. M. Khoshgoftaar, "Survey on categorical data for neural networks," *Journal of Big Data*, vol. 7, no. 1, pp. 1–41, 2020.

[23] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT Press, 2009.

[24] H. Scheffe, *The analysis of variance*. John Wiley & Sons, 1999, vol. 72.

[25] D. C. Montgomery and G. C. Runger, *Applied statistics and probability for engineers*. John Wiley & Sons, 2010.