Machine learning in calculating speeds in a railway sorting yard

A N Shabelnikov¹, I A Olgeizer¹ and A V Sukhanov¹

¹JSC “NIIAS” Rostov Branch, 344038 Russia, Rostov-on-Don, Lenina str. 44/13
E-mail: a.suhanov@rfniias.ru

Abstract. The main step in the process of disbanding railway rolling stock at the marshalling yard is the regulation of the speed of cars that slide freely from the hump, which in most cases is carried out thanks to the brake positions installed on the hump yard in several groups. The paper considers the solution of the problem of determining the optimal exit speed of a group of cars (trailers) target braking positions on the tracks of the sorting yard, since the efficiency of the sorting process in terms of optimal accumulation on the cars on the tracks of the sorting yard and its safety depends on the correctness of the calculation of this speed in terms of compliance with the permissible speed of collision with the cars on the tracks. In the proposed study, an analogy is drawn between the calculation of the optimal exit speed from the brake position and the compilation of nonlinear multiple regression. The classic and modern machine learning algorithms to build regression models were analyzed. The most suitable algorithm was identified within the study. In conclusion, the results of the introduction of machine learning tools at a real automation facility for sorting processes are presented.

1. Introduction

The sorting process is the process of disbanding trains arriving at the station and forming new trains according to a pre-prepared dissolution program [1]. For its implementation, a hump yards of cargo sorting stations are used. The main elements of a hump yard are the moving part, the top part, the rolling-down part and the sorting bowl part.

To regulate the speed of the trailers on the hump yard, brake positions are used. The brake positions located on the rolling-down part are intended for wiring the groups of wagons (cut of cars) after the top of the hump and for regulating the movement of the cuts of cars in the railroad switch zone, yard (sighting) brake positions - to ensure the acceptable speed of impact of the cuts of cars in the yard [2].

Yard brake positions have the greatest influence on the efficiency of hump yards [3]. Since they are located directly in front of the sorting bowl part, inappropriate braking can bring to negative consequences, such as:

1. High speed of connection of the cut of cars to the cars standing on the sorting tracks, which reduces the safety of the sorting process and brings to the risk of cargo damage.
2. Stop of the cut of cars long before the back of the car in front, which leads to the formation of gaps (windows) between the cars on the sorting track and reduces the efficiency of the sorting process
Implemented and functioning to date in an integrated system for automated control of the sorting process [4] the algorithm for calculating the exit speed of the couplers from the yard brake position is based on the data on the slopes of the longitudinal profile of the marshalling yard tracks, the physical parameters of the cars (weight, number of cars, restrictions on the type of cargo and rolling stock), as well as on current weather conditions. The algorithm uses manual adjustment tables. The need for manual adjustment of configuration tables was due to the low accuracy and reliability of the outdoor track filling control devices. Manual adjustment of configuration tables leads to a restriction from above on the dimension of these tables, which significantly reduces the number of parameters taken into account.

The ubiquitous transition to the use of the automated operating mode of hump yard control systems at stations equipped with automated control of the sorting process turns them from automated to fully automatic systems [5].

This led to the often-insufficient accuracy under the new conditions for calculating the speed of rolling stock, so the task of accurately calculating the exit speed of cars from the third braking position to the sorting bowl part is very urgent. In a mathematical formulation, this problem sounds like a task of regression analysis of the magnitude of the output speed of a cut from the parameters affecting it.

In the framework of this work, a study was made of the applicability of machine learning algorithms to the regression problem of calculating the exit speed from the third braking position based on the set of influencing parameters. For analysis, we used popular basic machine learning methods [6], such as linear regression, the support vectors method, and also known non-linear analysis methods, such as the k-nearest neighbors’ method, decision trees, random forest, gradient boosting, a multilayer neural network.

2. Used data
The data used in this study are information from the database of complex automated control system of the sorting process for four months (from October 2019 to January 2020) on the odd automated
marshalling yard hump of the Kinel station of the Kuibyshev railway. The total number of observations is 100000.

Hereinafter, we will consider optimal such a release speed of the release from the third braking position \( v \), at which the speed of its collision with the cars in the sorting bowl part will be at least 3 km/h and not more than 5.4 km/h.

Table 1 shows the variables used to build the regression model.

| No. | Parameter                                 | Unit of measurement |
|-----|------------------------------------------|---------------------|
| 1   | Cut of cars weight, \( x_1 \)           | t                   |
| 2   | The number of cars in the cut, \( x_2 \) | number of wagons    |
| 3   | Collision speed of a cut with cars in a yard, \( x_3 \) | m/s                 |
| 4   | The distance to the cut of cars in front of the cut at the time of exit from the rolling-down part, \( x_4 \) | m                   |
| 5   | Weather factor*, \( x_5 \)             | %                   |
| 6   | Air temperature, \( x_6 \)             | °C                  |
| 7   | Atmosphere pressure, \( x_7 \)         | mm of mercury       |
| 8   | Air humidity, \( x_8 \)                | %                   |
| 9   | Wind speed, longitudinal component, \( x_9 \) | m/s                 |
| 10  | Wind angle, \( x_{10} \)              | °                    |

*The coefficient is the data on current weather conditions combined by the Lagrange interpolation polynomial in one parameter according to the weather station [4].

3. Interpretation of the problem in the theory of machine learning

Machine learning or learning by precedent is an extensive class of mathematical algorithms, a characteristic feature of which is the memorization of general laws according to particular empirical data [7].

Memorization is a solution to the optimization problem, during which the minimization of some function called the loss function and characterizing how much the value calculated by the algorithm differs from the real empirical value.

Machine learning allows you to identify non-trivial patterns that cannot be described analytically.

Let us formulate the problem in the language of machine learning.

Many objects \( X = \{ x \} \), many classes \( Y = \{ y \} \) are set, and there is an objective function \( y: X \rightarrow Y \), whose values \( y_i = y(x_i) \) known only on a finite subset of objects \( \{ x_1, \ldots, x_n \} \subset X \) of some power \( n \). «Object-Class» Couples (\( x, y \)) are called precedents. Set of couples \( \{ x_i, y_i \}_{i=1}^n \) is called the training set.

The task is to find an algorithm that is capable to reestablish \( y \) according to \( x \in X \).

In our case \( x = [x_1, x_2, \ldots, x_{10}] \) (table 1), \( y = [v] \).

4. Research Methods

4.1. Multidimensional linear regression

Currently, multidimensional linear regression is often used to confirm that the data has some dependence, and not to find the dependence itself.

The following mathematical model is called multidimensional linear regression:

\[
\mathbf{f}(\mathbf{x}) = \sum_i a_i x_i , \tag{1}
\]

Where \( a_i \) – the weights of each i-th feature of the object \( \mathbf{x} \).
This model is the classic way to describe dependencies. Here, the mean square error $E$ is used as a loss function for all objects $x$:

$$E = \sum (f(x) - y)^2$$  \hspace{1cm} (2)

4.2. Support Vectors Method

Any regression task can be reduced to a classification problem if the quantitative attribute is converted to a nominal one using a threshold value as a criterion for classifying one or another class. In this case, a separating hyperplane is constructed in the attribute space, which defines all sets of threshold values for the attributes. If the observation lies far from the dividing hyperplane, the classification is certain. However, if the point is located on the separating surface, the classification is inaccurate and may contain errors.

The support vector method is a mathematical realization of the idea of maximizing the distance from each object to the separating hyperplane for a more confident classification. The dividing hyperplane can be both linear and nonlinear. A nonlinear hyperplane is more suitable. In this case, the loss function takes the form:

$$E = \frac{1}{2} ||w||^2 + c \sum (1 - M_i(w, b))$$  \hspace{1cm} (3)

where $w$ - dividing hyperplane,

$M_i(w, b)$ - distance (indent) from the hyperplane to each $i$-th object,

$c$ and $b$ – constants.

4.3. K-nearest-neighbors method

The $k$-nearest neighbors method is a metric method based on determining the similarity between objects by calculating some function of the distance between objects – a metric.

If the distance between objects is acceptable based on some threshold function, one class is assigned to them. As a metric we will use the Euclidean distance between objects $x_1 = \{x_{1i}\}$ and $x_2 = \{x_{2i}\}$:

$$D(x_1, x_2) = \sqrt{\sum (x_{1i} - x_{2i})^2}$$  \hspace{1cm} (4)

Learning the model of $k$-nearest neighbors consists in recalculating the distances from all objects to the desired one and in selecting the $k$ nearest objects. Which class the majority of neighbors belongs to, such a class is assigned to the considered object.

4.4. Algorithms based on ensemble models

Ensemble models - models that use the composition of weaker models to achieve better approximation accuracy and better solution stability. In this study, two models are taken: random forest and gradient boosting.

4.4.1. Random forest

A random forest is an ensemble of independent decision trees, each of which seeks to reduce the state entropy:

$$S = -\sum p_i \log_2 p_i$$  \hspace{1cm} (5)

where $p_i$ – probabilities of finding the system in $i$-th condition.

The decrease in entropy occurs by dividing into groups by characteristics with verification of the growth of information for each characteristic $Q$:
\[
IG(Q) = S_0 - \sum_{i=1}^{q} p_i S_i
\]

where \(S_0\) – state entropy before splitting,
\(q\) – number of splits,
\(p_i\) – probability of finding an object in the \(i\)-th group,
\(S_i\) – state entropy in the \(i\)-th splitting.

If the tree fails (calculation accuracy is below a certain threshold value), it is discarded from the overall composition. One of the positive properties of a random forest is that with an increase in the number of trees in the ensemble, the accuracy of calculations increases. The composition method used to build a random forest is called bagging.

4.4.2. **Gradient Boosting**

Gradient boosting, unlike a random forest, builds trees of small depth, while the construction is iterative, and each subsequent tree takes into account the errors of the previous one.

If the exclusion of any one tree from a random forest makes approximately the same contribution to the overall accuracy, then with gradient boosting this is not so, and the exclusion of the first tree makes the maximum contribution to the overall error. Boosting is called gradient, because it is an optimization method based on modifications of gradient descent.

4.5. **Multilayer neural network**

An artificial multilayer neural network is inherently the same regression model based on optimization of the mean square error (2), where \(f(x)\) – nonlinear dependence (function activation of the neuron of the output layer) of the form:

\[
f(x) = f \left( \sum_i w_i g_i(x) \right)
\]

where \(g_i(\cdot)\) – activation function of the \(i\)-th neuron of the last hidden layer
\(w_i\) – weight of \(i\)-th function.

Function \(g_i(\cdot)\) with respect to the previous hidden layer, it is also calculated by the formula 7. In the case of the activation function of the neurons of the input layer, value of \(i\)-th feature of the object is used instead of \(g_i(\cdot)\).

Most often, a sigmoidal activation function is used as an activation function:

\[
f(u) = \frac{1}{1 + e^{-u}}
\]

The task of minimizing the error, which is the essence of all training, comes down to differentiating the error function and finding the minima. Traditionally, this problem is solved by the gradient descent method. [8].

5. **Results of the study**

Since a practical comparison of the existing and considered algorithms at a working facility without comprehensive verification and debugging is impossible due to security considerations in the process of disbanding compositions, the following evaluation approach was used for comparison.

In order to ensure the adequacy of the comparison of algorithms, only those cases were considered in which the estimated exit speed from the yard brake position according to the existing algorithm \(v_{ex}^e\) equal to the real exit speed of the yard brake position \(v_{ex}^r\).

\(V_{ex}\) considered accurately calculated if \(x_j \in [3;5.4] \text{ km} / \text{h}\).

Let us designate the estimated exit speed according to the considered algorithm (one of the algorithms of the section 4) as \(v_{calc}\).

Let the \(\alpha\) – be an accuracy indicator for \(v_{calc}\). Then, based on the foregoing, \(\alpha\) is calculated as:
\[ \alpha = \begin{cases} 
1, & \text{if } (x_3 \in [3; 5.4] \wedge v_{\text{calc}} = v_{\text{ex}}) \\
\vee \quad (x_3 > 5.4 \wedge v_{\text{calc}} \in [v_{\text{ex}} - \Delta v; v_{\text{ex}}]) \\\n\vee \quad (x_3 < 3 \wedge v_{\text{calc}} \in [v_{\text{ex}}; v_{\text{ex}} + \Delta v]) \\
0, & \text{otherwise}
\end{cases} \]  

(9)

where \( \Delta v \) – margin of variation between \( v_{\text{calc}} \) and \( v_{\text{ex}} \), in which comparison is still adequate.

6. Experiment Results

To build a random forest (4.4.1) and a gradient boosting (4.4.2) 3000 trees were used. Neural network (4.5) consisted of an input layer, an output layer and two hidden layers. To implement the algorithms programming language Python 3 was used.

The ratio of the training sample to the test one was 3:1.

By a numerical characteristic of the accuracy of the algorithm, we take the ratio of accurate predictions \( m \) to the total number of measurements \( n \), i.e.:

\[ p = \frac{m}{n} \]  

(10)

The results of experiments to test various machine learning methods based on formula (10) are presented in table 2.

| Algorithm                        | Accuracy, \( p \) |
|----------------------------------|-------------------|
| Existing algorithm               | 0.38              |
| Linear regression (4.1)          | 0.30              |
| Support Vectors Method (4.2)     | 0.35              |
| Method of k-nearest neighbors (4.3) | 0.40         |
| Random forest (4.4.1)            | 0.45              |
| Gradient Boosting (4.4.2)        | 0.49              |
| Multilayer neural network (4.5)  | 0.50              |

As can be seen from table 2, the neural network, random forest, and gradient boosting are the algorithms with the minimum dispersion value and the maximum prediction accuracy.

It should also be noted that the training sample used is quite small - large data on the operation of the hump yard for 4 months accumulated by the integrated system for automated control of the sorting process do not take into account the seasonality factor. In this case, training of a neural network quickly reaches a certain plateau, which can be illustrated on the graph of the loss function (Figure 2) from the number of the training epoch.

![Figure 2. The dependence of the mean squared error on the epoch number.](image-url)
The graph in Figure 2 shows that the value of the error of the neural network reaches a “plateau” after the 10th epoch, a further decrease in the error is possible with an increase in the training sample.

7. Discussion of results and conclusion

The paper illustrates the implementation of basic machine learning methods for calculating the speed of exit of railway couplings from park brake positions of a hump yard, taking into account the conditions for a safe connection with the tail of the forward train. Based on the experiments, it is shown that the presented calculations are an urgent task for further development and require the introduction of marshalling yards automation at real facilities.

In fact, it is possible to use the results obtained to calculate the exit speed of the couplings from the park brake positions on the marshalling yard railway, in order to achieve the optimal joining speed between the couplings during the formation of new trains.

Moreover, in addition to increasing the quality of forecasting and accumulation of compositions, the use of the proposed algorithms provides a number of additional advantages:

- formation of setup tables based on the results of training the neural network in automatic mode;
- a multiple increase in the criteria taken into account for rolling stock;
- a sharp decrease in labor costs for processing and adjusting settings; the formation of estimated exit speeds from park brake positions.

The results obtained are planned to be implemented in the integrated system for automated control of the sorting process this year.

Acknowledgment

This work was supported by the Russian Foundation for Basic Research (projects # 19-07-00263, # 19-07-00195 and # 20-07-00100).

References

[1] Ivanchenko V N, Lyashchenko A M 2014 Principles of construction and hardware-software implementation of the subsystem of intellectual support for train disbanding processes News of the Southern Federal University Engineering (6 (155))

[2] Shabelnikov A N, Olgheiser I A, Rogov S A 2015 Innovative technology of smooth control of brake means Automation, communication, informatics (3), pp 15–17

[3] Sokolov V N, Rogov S A 2010 Control of rolling speed and aimed braking of couplings. Rail transport (8), 41–42

[4] Lyabakh N Nø Olgeizer I A 2009 Algorithm and methodology for taking into account climatic effects on the sorting process. Bulletin of the Rostov State University of Railway Engineering (1), pp 69–72

[5] Shabelnikov A N, Sokolov V N 2017 Integrated system for automated control of the sorting process - a new direction in the automation of hump yards. Automation, communication, informatics (8), pp 2–4

[6] Alpaydin E 2020 Introduction to machine learning. MIT press

[7] Ruder S 2016 An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747

[8] Muller A C, Guido S 2016 Introduction to machine learning with Python: a guide for data scientists. O'Reilly Media, Inc

[9] Raschka S, Mirjalili V 2019 Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2. Packt Publishing Ltd