A Machine Learning Application in Brazilian Railway Crew Rostering

Rael Fonseca Andretto
Graduate Program in Computational Modeling
Universidade Federal de Juiz de Fora
Juiz de Fora, Brazil
rael.andretto@engenharia.ufjf.br

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

***Abstract*—A large portion of railway expenses is in the workforce's payment. Although there are specialized professionals whose function is to appoint the drivers to the trains economically, many uncertainties exist at the time of the decision, leading many drivers to carry out idle cycles. Machine learning techniques are ideal for generalizing new scenarios from a training database and can be applied to reduce uncertainties in many problems. The literature shows that there are few machine learning applications in crew scheduling and rostering, especially in railways. In this study, applied at a Brazilian railway operator, in the first step, exploratory data analysis techniques are used together with the rules generated by a decision tree to create and apply guidelines to reduce idle crew cycles. In the second step, five machine learning algorithms are evaluated to automate and improve the process: neural network, support vector machine, decision tree, random forest, and the Autonomous Learning Multi-Model System (ALMMo). Although the first step got acceptable results and has been applied in the company, the machine learning models improved the result, showing an accuracy above 86% on average, which meets all service levels established by the company. Finally, the decision tree, the random forest, and the ALMMo were considered suitable solutions for application in the company due to their performances and characteristics.**

***Index Terms*—railway, machine learning, crew scheduling**

I. Introduction

Having only a tiny portion of the Brazilian transport matrix, railway companies seek to reduce costs and optimize processes to become increasingly competitive. Within operating costs, the qualified labor expenses stand out, which include payment for unproductive hours, that is, hours waiting for the train's arrival, overtime, displacement, food, and hotel costs [1].

In addition, inadequate planning can lead to not having enough train drivers to meet the demand. The need for drivers is a problem that affects rail operators all over the world. In 2019, it was reported in The Times that over 35,000 trains were canceled in the last six years due to a lack of crew [2]. In this context, reducing idle cycles for train drivers directly generates financial savings for the company and can also avoid the lack of train conductors since, if dismissed, the driver can be available again in his depot earlier. The forecast of the driver's actual utilization in his cycle can be performed through computational intelligence models.

Although the fundamentals of machine learning were developed in the last century, it was only with the increase in the processing capacity of computers that these models gained wide application. Currently, computational intelligence is successfully used in several sectors, such as medicine, law, games, self-driving cars, art creation, and transport. On the railroad, the use of machine learning is very concentrated in the area of maintenance and inspection, with its utilization in crew scheduling and rostering being a field still under development, with a lot of opportunity for further studies [3].

The main objective of this work is to develop a solution, based on computational intelligence techniques, to reduce the number of drivers allocated in a section of a Brazilian railroad. In the study, guidelines will be created for the dismissal of surplus drivers, and the performance of machine learning algorithms will be evaluated for application in the company.

The remainder of this paper is structured as follows: literature review in Section II, problem formulation in Section III, the theoretical basis of the study in Section IV, then the experiments and results are discussed in Section V, and finally, Section VI concludes the paper.

II. RELATED WORK

The Crew Scheduling Problem (CSP) [4] is under constant research in the scientific community. However, these applications are often restricted to optimization techniques, such as linear programming or even metaheuristics.

In [5], the authors describe the main concepts of crew rostering:

- Task: A one-time job, such as taking a train from Yard A to Yard B.
- Duty: A set of tasks to be done in a cycle.
- Depot: Location where the service starts and ends. Ideally, the driver's location should also be his city of residence.
- Rostering: To assign a task to a driver.
- Over-Covered Task: Allocate more than one driver for the same task without being necessary. Despite initially

This work has been supported by the Brazilian agencies (i) National Council for Scientific and Technological Development (CNPq), Grants no. 310625/2022-0; (ii) Coordination for the Improvement of Higher Educational Personnel (CAPES); (iii) Foundation for Research of the State of Minas Gerais (FAPEMIG in Portuguese), Grant no. APQ-02922-18; (iv) MRS Logística S.A., and (v) Stemis Serviços de Tecnologia Ltda.

seems expensive, the over-covered train may be needed to reduce the overall cost.

Machine learning techniques are often used to solve railway problems. But, as described by Tang *et al.* [3], the researches are more concentrated on maintenance and inspection. After extensive research, the only publication with an application of machine learning in Railway Crew Scheduling Problem (RCSP) found was the one developed by Gattermann-Itschert, Poreschack, and Thonemann [6]. The authors model a random forest to predict if the planner would consider a duty good or bad to integrate it into the optimization algorithm later. With this approach, they reduced the number of adjustments made to the crew scheduling provided by software. Using computational intelligence techniques allied to the CSP is a little more common in air transportation as seen in [7] and [4]. Both utilized neural networks to reduce the computation cost of their CSPs.

Rodrigues [1] used the simplex method of linear programming to generate the most economical crew allocation on the Brazilian railway path from the city of Juiz de Fora to Lafaiete city, thus automating the crew rostering process. In this study, it was not evaluated whether a spare train driver should be dismissed or over-allocated.

III. PROBLEM FORMULATION

In the Brazilian railway company analyzed, MRS Logística S.A., as described in [1], the assignment of a train to a specific driver is only done in the daily crew schedule, also called ultra-short-term planning [5]. This allocation is made with a range of only the next twelve hours due to the ore trains not having a fixed timetable to run, performing a cycle of loading and unloading in sequence depending only on demand and the performance of the terminals. Twelve hours is considered the maximum range of confidence for forecasting train arrivals.

This research focuses on implementing computational intelligence in the assignment of railway crew with an ultra-short-term view in the Brazilian railway company. This allocation is carried out by the position of the “crew scheduler”. The rostering input data is the train arrival forecast in the next twelve hours, the arrival time of drivers in the analyzed depot, and their qualifications. More specifically, this paper will focus on the allocation of drivers in two yards: the FDT yard, located in the city of Juiz de Fora, and the FJC yard, located in the city of São Brás do Suaçui, both in the Brazilian ore carousel.

According to Rodrigues [1], the train driver from Juiz de Fora has only one possible type of duty: to drive a train from FDT to the city of Lafaiete, heading to the rest time regulated by law lasting 10 hours, after this time, he will be driven by car to FJC, where he will conduct a train to the city of Bom Jardim, returning by car to his depot in Juiz de Fora, leaving then for the day off. For some operational reasons, some drivers can go from FDT to Lafaiete city and not make the second part of the duty, but those are exceptions. Considering unproductive times, in transport, on the train, and at rest, it is likely that the driver will be available at FJC on the day after the start of duty. The FDT driver’s cycle is illustrated in Fig. 1.

The general rules for assigning a train to a driver are:

- A train driver should only be allocated to a task that he can fulfill within the working hours regulated by law.
- Train delay due to lack of driver must be avoided.
- The driver’s waiting time (idle time) must be minimized as long as it does not generate major impacts on train circulation.

In a scenario with a driver in surplus, the driver’s allocation according to the FIFO (First-in First out) rule would occur according to the example in Table I.

In this FIFO scenario, the drivers are waiting long for the trains, and the last employee is being left over. Considering the reduction of waiting times for drivers and without causing train delays, it will be up to the crew scheduler to adjust the rostering to become more economical, as seen in Table II, in which trains have been assigned to a driver more economically. The remaining driver may be dismissed from duty or allocated on the next train, which will be over-covered. The advantage of the dismissal is the direct economy for the company of not having to spend on unproductive hours, car travels, and hotel for this driver. However, if this driver does not take the train from FDT to Lafaiete city, he will not be at FJC the next day to drive trains, which can cause a lack of drivers at FJC, depending on the daily demand. For this reason, even without the need in the first train, the driver is generally not dismissed, and the train goes over-covered. But this driver may also be idle on the way back, returning in a train over-covered, performing an entire cycle completely idle.

Considering that the driver is only available at FJC the day after the start of his duty at FDT, it is clear that when this driver is assigned to the first train, given that the rostering range is of only twelve hours, it is not possible to know at this first moment if he will be needed at FJC. It is not known which train or even if there will be a train for his return from FJC. It was adopted as a rule in the company’s crew rostering that the surplus driver should always be assigned to a train since, due to the lack of information, it was better to have

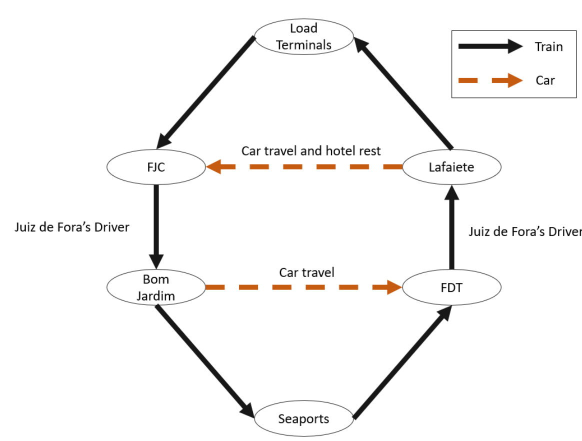

Fig. 1. FDT Driver’s Cycle.

TABLE I
EXAMPLE OF A CREW SCHEDULE WITH A SURPLUS DRIVER.

Driver	Driver's Arrival Time	Train	Train's Arrival Time
Driver 1	8:00	Train A	9:00
Driver 2	10:00	Train B	12:00
Driver 3	16:00	Train C	18:00
Driver 4	17:00	Train D	19:30
Driver 5	19:00	_	_

TABLE II
EXAMPLE OF AN OPTIMIZED CREW SCHEDULE WITH A SURPLUS DRIVER.

Driver	Driver's Arrival Time	Train	Train's Arrival Time
Driver 1	8:00	Train A	9:00
Driver 2	10:00	Train B	12:00
Driver 3	16:00	_	_
Driver 4	17:00	Train C	18:00
Driver 5	19:00	Train D	19:30

surplus drivers at FJC than shortages. Furthermore, this lack of information prevents an optimization-only solution from being performed. The arrival of train drivers at FJC is directly linked to the scenario of trains along the FDT line. To illustrate this premise, the view of the train graph will be used, in which the yards passed by the train are on the y-axis and the time on the x-axis. In Fig. 2, there is a big gap in train arrivals in Juiz de Fora (FDT), meaning there will be five hours without train drivers getting trains for Lafaiete, causing a five-hour gap without drivers arriving at FJC. It would be essential to put more than one driver on the train that precedes the gap in circulation, if there are surplus crew, because when these drivers in surplus arrive at FJC, they will fill the future gap in drivers' arrival.

In the case of Fig. 2, it seems logical to over-cover a train, as FJC will not receive train drivers for many hours. However, some scenarios are not so simple to understand. For example, in Fig. 3, there is not a significant gap in train circulation, but besides having fewer trains in the day, these are more spaced from each other. In comparison, the daily scenario of FDT varies greatly, according to Fig. 2 and to Fig. 3, and the scenario of FJC tends to have a constant flow of trains without significant variations.

As noticed, there are many different circulation scenarios

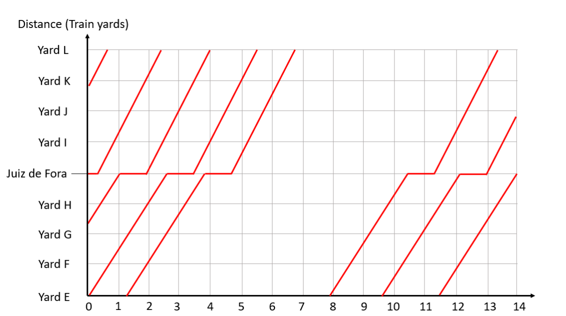

Fig. 2. Scenario with a Great Gap in Circulation.

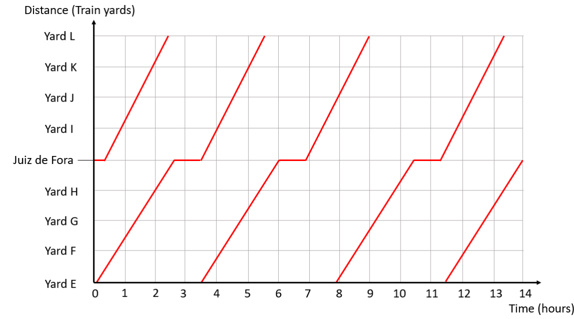

Fig. 3. Scenario with Few Trains.

at FDT, and it needs to be clarified when over-covering a train is necessary. The problem to be studied is reducing train drivers' idle cycles, in the yards FDT and FJC, by forecasting the need to over-cover a train in the ultra-short-term crew rostering. Classification machine learning models will be utilized to provide information on the real need of a surplus driver after the crew scheduling optimization has been performed, either manually as is currently done by the company, or as solved by Rodrigues [1]. That is, after the crew scheduling of a depot concludes that the driver is not needed on the first trip, the machine learning model will predict if the driver will be needed on the second trip. If not needed on both trips, he can be dismissed from this duty, saving costs for the company. As shown previously, the related works focused on reducing the computational cost of the optimization, while this methodology utilizes machine learning to reduce even more the economic costs of the operation after the optimization is done.

IV. THEORETICAL FOUNDATION

The decision tree is a learning model that consists of the subdivision of the space of instances according to the variable's value that represents the best reduction in the impurity measure [8]. One of the significant advantages of the decision tree is that it is possible to understand the paths that led the model to classify data in a specific category.

According to Breiman [9], the random forest is a classifier consisting of a collection of k decision trees $h(x, \theta_k)$, where θ_k are random vectors that control the parameters of the trees.

The support vector machine (SVM) is a machine learning model that searches for the hyperplane that increases the margin of separation between the classes [10]. To predict data that are not linearly separable, kernel functions are used to map the data into a new dimensional space.

The feedforward neural network is a nonlinear model composed of connected layers of artificial neurons. The learning of a feedforward network is done by updating the weights of each connection through the propagation of the partial derivative of the error function from the last layer to the first one [11].

The *Autonomous Learning Multi-Model System*, or ALMMo [12], can be understood as a grouping of systems based on AnYa type fuzzy rules. This nonparametric and evolutionary model makes no assumptions about the data, does not need to adjust hyperparameters, and is updated with each new input without retraining the model on the entire database.

The Wilcoxon test [13] is a nonparametric statistical test based on ranks and can be used on dependent data, such as, for example, comparing data before and after an event or comparing results of two algorithms on the same data.

V. EXPERIMENTAL RESULTS

A two-step study was developed with the objective of resource-saving by predicting when to dismiss a driver when he is a surplus in the first part of his cycle in the FDT yard. An initial data analysis study was carried out in 2019 and machine learning models were evaluated to get better solutions in 2022.

The machine learning models were run in the Python 3.7 language and using the Scikit-Learn [14] library, with the exception of ALMMo. The computer utilized has an AMD Ryzen 7 5800X 8-Core 3.80 GHz processor, 16.0 GB of RAM, and Windows 11 Pro.

The database was built from several train scenarios between the years 2018 and 2022, in which each sample represents an over-covered train in Juiz de Fora city. The data were extracted from the company's train circulation software and from the crew allocation system. The variables were chosen by the "crew schedulers" according to what could have a relationship with the drivers' idle cycles in the daily work. The following variables were selected:

- T_Drivers: total number of drivers on the train;
- N_Drivers: total number of drivers on the train who will be at FJC in the future;
- N_Trains_Before: numerical variable representing the number of trains that passed before the over-covered train in a four-hour range. This range was defined due to the Brazilian railway driver's shift limit being of 12 hours. That is, if he performs an eight-hour job, he can be unproductive for a maximum of four hours. This variable and the next one seek to evaluate how much the scenario was saturated with trains and was separated into "before" and "after", due to the practical experience leading the crew scheduler to believe that the scenery after the train would have greater influence in idle cycles than the previous one;
- N_Trains_After: similar to the previous variable, but to measure the number of trains after the over-covered one.
- Headway_Before: numerical variable representing the number of hours that passed between the last train and the over-covered one.
- Headway_After: similar to the previous variable, but with the headway of the over-covered train and the next one.
- Train_Type: categorical variable that represents the type of the train, whether it was iron ore or general cargo.
- Weekday: day of the week (categorical).
- Time: morning, afternoon, or night (categorical).
- T: the target variable. "OK", if the surplus driver was necessary and "N" otherwise.

The database contains 997 samples, 521 correctly over-covered trains, and 476 with full idle cycles. The data was considered as balanced. Table III represents a small example of the database.

A. 2019's EXPERIMENT

In 2019, as there needed to be more information about the real need of the drivers in the return stage of their cycle, all drivers in surplus in the FDT schedule were not dismissed. Introducing new technologies to change processes, in addition to obtaining valuable data, can be a very time-consuming process to implement in the corporate environment. Aiming at a palliative and quick-to-implement solution to the aforementioned problem of idle cycles, a study of data analysis was carried out and a construction of a decision tree was proposed to generate explicit rules to be followed. This initial study was simplified due to the need to have a quick solution to the problem, even if not ideal. The decision tree was chosen because it is possible to know its internal rules and, therefore, apply these rules in the operation, while there was no environment in which the direct application of machine learning was possible.

In this first stage, with the base still under construction, only eighty samples and the variables T_Drivers, N_Drivers, Headway_Before, Headway_After, and T were used. To compare the linear correlations, the values of the target variable T were converted to a number, with "OK" being transformed into 0 and "N" into 1. So a significant and positive linear correlation between a variable and the target means that the greater this variable is, the greater the probability of an idle cycle to occur.

The correlations between the variables and the target (T) are in Table IV. From it, it can be noticed that the gap after the over-covered train has a higher correlation, in absolute value, than the past one. This high correlation can be explained because when over-covering before the gap in circulation, the surplus driver will be available at FJC when there will be no other drivers there.

For the tree's construction, it was defined to have a maximum of two levels below the origin. After all, a larger tree would generate many rules to be analyzed by the crew scheduler, which could lead to confusion, slow decision-making, and even difficulty in implementing the process. The

TABLE III
DATABASE EXAMPLE.

T Driv.	*N Driv.*	*Trains Before*	*Trains After*	*Head. Before*	*Head. After*	*Train Type*	*Day*	*Time*	*T*
2	2	3	2	0,75	1,25	HH	*Wed*	*Morn.*	*N*
2	1	1	3	0,50	1,50	HH	*Thu*	*Morn.*	*OK*
2	2	2	4	2,00	0,50	HH	*Thu*	*After.*	*OK*
2	2	1	5	0,50	0,50	GC	*Thu*	*Night*	*N*
2	2	5	4	0,50	1,00	HH	*Fri*	*Night*	*N*

TABLE IV
LINEAR CORRELATIONS.

	T_Drivers	**N_Drivers**	**Headway Before**	**Headway After**
Target	0.14	0.2	0.13	-0.42

database was divided between train and test in the proportion of 80% and 20%, respectively, and the code was executed 100 times. The classification metrics [15] are displayed in Table V.

As can be noticed, the accuracy of the model was around 70%. This is likely because few data was collected, and few variables were evaluated for the model. The gap and number of drivers may be one of many factors to be taken into account for performing an idle cycle. In addition, with a small base, many scenarios may have yet to be considered.

Analyzing the decision tree generated by the data in Fig. 4, it can be noticed that it classifies all samples with Headway_After less than 1.795 hours as "idle cycle". Samples with Headway_After greater than 1.795, if with three or more drivers traveling (N_Drivers) will also be classified as "idle cycle". Classification as a "useful cycle" only occurs with a Headway_After greater than 1.795 and two or fewer N_Drivers on the train.

The tree classification rules have been simplified in the following guideline: To only over-cover a train if the gap after this train is greater than two hours. This guideline became known as the "Over-Covering Rule" and began to be applied in the crew schedule from the second quarter of 2019. Fig. 5 shows the number of train drivers who performed idle cycles, per quarter, from 2018 to 2022. It can be noticed that after using the "Over-Covering Rule", the number of idle cycles drops significantly. At the company's request, the number of train drivers was represented as a percentage of the worst result in the first quarter of 2018. The target value desired by the company for idle cycles to be performed is displayed in red, which is also reduced each year.

It can be noticed that following the rule made idle cycles drop more than half the initial value. In addition, at the end of 2019 and in 2020, the values were well below the maximum desired by the company. As of 2021, there was a slight increase in idle cycles performed. Since the creation of the guideline, there needed to be an updated analysis of the data or updates to the rules. Over time the policy became less efficient. Even so, it maintained a very appropriate level of reduction compared to the quarters in which the guideline was not yet applied.

TABLE V
DECISION TREE RESULTS.

	Accuracy	Precision	Recall	F1 Score
Tree	0.70 ± 0.10	0.68 ± 0.10	0.70 ± 0.18	0.67 ± 0.10

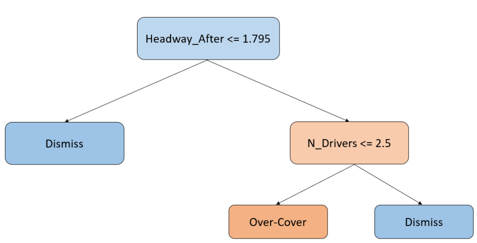

Fig. 4. Decision Tree that Generated the "Over-Covering Rule".

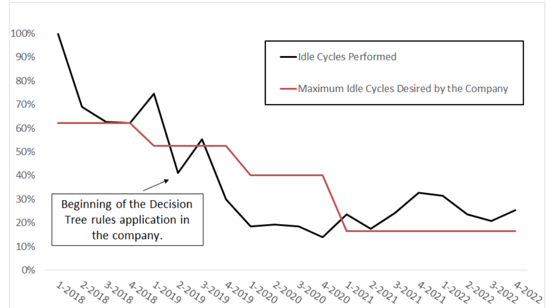

Fig. 5. Idle Cycles vs Reduction Expected by the Company.

B. 2022's EXPERIMENT

At the end of 2022, five machine learning models were evaluated in the complete database to improve the previous results: neural network, SVM, decision tree, random forest, and ALMMo.

The choice of models is mainly due to their characteristics, such as the understanding of their internal rules, in the case of the decision tree, being evolutionary for ALMMo, and being algorithms of high predictive power with many applications in the literature, such as the neural network, SVM and random forest. To simplify the terminology, the neural network will also be identified as "MLP", the decision tree as "Tree", and the random forest as "Rnd."

The categorical variables were converted with the Ordinal Encoder from the Scikit-Learn [14] library, and all data were scaled from 0 to 1, because learning methods such as SVM and MLP are sensitive to the scale of the variables.

To define the hyperparameters of each algorithm, the Grid Search [16] was used with cross-validation of 5 folds. In Table VI are the selected hyperparameters. For ALMMo, as this model is nonparametric, there was no hyperparameter adjustment step.

TABLE VI
SELECTED HYPERPARAMETERS.

Model	Parameters	Values
Neural Network	Hidden layers	(3,2)
	Hidden layers' activation function	Tanh
	Solver	Adam
	Alpha (regularization)	0,001
	Learning rate	Constant
	Initial learning rate	0,001
SVM	C (regularization)	0,01
	Kernel	'poly'
	Kernel function coefficient	10
	Kernel function independent coefficient	10
	Kernel's polinomial degree	3
Decision Tree	Division criteria	Entropy
	Maximum number of variables to perform division	5
	Max tree depth	5
	Minimum number of samples to perform division	10
	Minimum number of samples in leaf	2
	Maximum number of leafs	9
Random Forest	Number of trees in the ensemble	40
	Division criteria	Gini
	Maximum number of variables to perform division	6
	Max tree depth	8
	Minimum number of samples to perform division	2
	Minimum number of samples in leaf	3
	Bootstrap	True

The database was divided into 80% for training and 20% for testing and trained in the five mentioned learning models. The experiment was repeated 100 times, randomly reordering the base. Afterward, the mean and standard deviation were computed for accuracy, precision, recall, and F1 Score. The results were compiled in Table VII

The nonparametric Wilcoxon test was used to verify the statistical significance of the results. The null hypothesis represents that the data come from the same population, and rejecting it means that the data come from different distributions. The statistical significance used was 5%. Table VIII contains the p-values obtained, rounded to the fourth decimal place when comparing the results of the models' accuracies. It can be seen that the model that got the best result, the random forest, rejected the null hypothesis against all other models. However, the MLP did not obtain significance when compared to the SVM, and the ALMMo did not get significance when compared to the tree. So, it is not possible to say that the MLP performed better than the SVM and neither the ALMMo concerning the tree.

Fig. 6 displays the number of idle events that the random forest would generate, showing a noticeable gain over the current "Over-Covering Rule". It can be noticed that with the use of the model, the results are well below the maximum value expected by the company, that is, reaching the desired results with ease.

The random forest model was the one that reached the best values in all metrics used. Therefore, it is the recommended model that will correspond to the best classification performance. However, considering that the tree and ALMMo models performed only 4% worse than the forest, a number that can be regarded as irrelevant in monthly crew savings, these models can also be recommended because they have other characteristics of interest. The decision tree is a model that allows its rules to be very explicit. So, it becomes a suitable model for the corporate environment, in which decision-making is better accepted when based on detailed rules. ALMMo has another essential characteristic for daily operations. As seen when applying the "Over-Covering Rule", it became obsolete over time. The same can be said for

TABLE VII
MODELS' PERFORMANCE (WITH STANDARD DEVIATION).

	Accuracy	Precision	Recall	F1 Score
ALMMo	0.86 ± 0.02	0.85 ± 0.03	0.86 ± 0.03	0.85 ± 0.02
MLP	0.88 ± 0.03	0.87 ± 0.04	0.87 ± 0.06	0.87 ± 0.04
SVM	0.88 ± 0.02	0.88 ± 0.03	0.87 ± 0.03	0.87 ± 0.02
Tree	0.86 ± 0.02	0.85 ± 0.04	0.86 ± 0.04	0.85 ± 0.02
Rnd	$\mathbf{0.90 \pm 0.02}$	$\mathbf{0.89 \pm 0.03}$	$\mathbf{0.91 \pm 0.03}$	$\mathbf{0.90 \pm 0.02}$

TABLE VIII
WILCOXON TEST P-VALUES FOR THE MODELS' ACCURACY.

	ALMMo	MLP	SVM	Tree	Rnd
ALMMo	-	0.0000	0.0000	0.0861	0.0000
MLP	0.0000	-	0.7629	0.0000	0.0000
SVM	0.0000	0.7629	-	0.0000	0.0000
Tree	0.0861	0.0000	0.0000	-	0.0000
Rnd	0.0000	0.0000	0.0000	0.0000	-

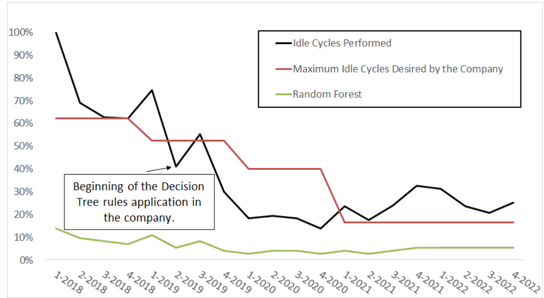

Fig. 6. Idle Cycles with Random Forest.

machine learning models. They must be retrained with more recent data occasionally to perform well. Then, the great advantage of ALMMo is that it is evolutionary. It does not need to be retrained in the entire database, only in the new data.

VI. CONCLUSION

With the development of this work, a machine learning solution was proposed to reduce the number of drivers performing idle cycles at a Brazilian railway company. Before the research, there were few dismissals in the company. An initial study with a significantly reduced database created an internal rule to only over-cover if there was a two-hour gap between trains. This rule was applied and generated promising results in the first years of application. Still, with changes in the scenario, it was realized that an alternative to this rigid guideline was necessary.

Random forest had the best performance in the classification metrics. However, the other models had at most a 4% lower result, thus not making such a relevant difference in the monthly crew savings. The company has three interesting options according to what best meets its future needs. The use of the random forest is recommended if the company prefers the best performance, even with small gains compared to other solutions. The decision tree can be ideal, as it generates better clarity of its internal rules, making it easier to apply in the corporate environment. Finally, ALMMo is still helpful in providing a solution that evolves without needing to be retrained on the entire database.

REFERENCES

[1] R. S. Rodrigues, "Alocação Eficiente da Tripulação da MRS Logística," Monography, IME, Rio de Janeiro, RJ, 2015.

[2] S. Dawson. "Rail crew shortages lead to 35,000 train cancellations." The Times. https://www.thetimes.co.uk/article/rail-crew-shortages-lead-to-35-000-train-cancellations-bwk05lmp2 (accessed May. 18,2022).

[3] R. Tang *et al.*, "A literature review of Artificial Intelligence applications in railway systems," Transportation Research Part C, vol. 140, May. 2022, doi: 10.1016/j.trc.2022.103679

[4] Y. Yaakoubi, F. Soumis, and S. Lacoste-Julien, "Machine learning in airline crew pairing to construct initial clusters for dynamic constraint aggregation," EURO Journal on Transportation and Logistics, vol. 9(4), pp. 227-251, Dec. 2020, doi: 10.1016/j.ejtl.2020.100020

[5] E. Abbink, D. Huisman, and L. Kroon, "Railway Crew Management," in Handbook of Optimization in the Railway Industry, R. Borndörfer, T. Klug, L. Lamorgese, C. Mannino, M. Reuther, and T. Schlechte, Eds., Cham, Switzerland: Springer, 2018, pp. 243-264.

[6] T. Gattermann-Itschert, L. Poreschack, and U. Thonemann, "Using machine learning to include planners' preferences in railway crew scheduling optimization," Transportation Science, in press. doi: 10.1287/trsc.2022.1196

[7] B. Gopalakrishnan and E. Johnson, "Airline crew scheduling: state-of-the-art," Annals of Operations Research, vol. 140, pp. 305-337, Nov. 2005, doi: 10.1007/s10479-005-3975-3

[8] L. Rokach and O. Maimon, "Decision Trees," in Data Mining and Knowledge Discovery Handbook, L. Rokach, O. Maimon, Eds., Boston, USA: Springer, 2005, pp. 165-192.

[9] L. Breiman "Random Forests," Machine Learning, vol. 45, pp. 5-32, Oct. 2001, doi: 10.1023/A:1010933404324

[10] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A Training Algorithm for Optimal Margin Classifiers," Proceedings of the Fifth Annual Workshop on Computational Learning Theory, pp. 144-152, Jul. 1992, doi: 10.1145/130385.130401

[11] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," Nature, vol. 323, pp. 533-536, October 1986.

[12] P. P. Angelov, X. Gu, and J. C. Principe, "Autonomous learning multi-model systems from data streams," IEEE Transactions on Fuzzy Systems, vol. 26(4), pp. 2213-2224, Nov. 2017, doi: 10.1109/TFUZZ.2017.2769039

[13] F. Wilcoxon, "Individual Comparisons by Ranking Methods," Biometrics Bulletin, vol. 1(6), pp. 80–83, Dec. 1945, doi: 10.2307/3001968

[14] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," The Journal of Machine Learning Research, vol. 12, pp. 2825–2830, Nov. 2011, doi: 10.5555/1953048.2078195

[15] M. Hossin and M.N. Sulaiman, "A review on evaluation metrics for data classification evaluations," International Journal of Data Mining & Knowledge Management Process, vol. 5, pp. 01-11, Mar. 2015, doi: 10.5121/ijdkp.2015.5201.

[16] T. Yu and H. Zhu, "Hyper-Parameter Optimization: A Review of Algorithms and Applications," in ArXiv , Mar. 2020. [Online]. Available: https://arxiv.org/abs/2003.05689 (accessed May. 18, 2022), doi: 10.48550/arXiv.2003.05689