Pineda-Jaramillo, J.; Martínez Fernández, P.; Villalba Sanchis, I.; Salvador Zuriaga, P.; Insa Franco, R. (2021). Predicting the traction power of metropolitan railway lines using different machine learning models. International Journal of Rail Transportation. 9(5):461-478. https://doi.org/10.1080/23248378.2020.1829513

Predicting the traction power of metropolitan railway lines using different machine learning models

Pineda-Jaramillo J.[a], Martínez-Fernández P.[b], Villalba-Sanchis I.[b], Salvador-Zuriaga P.[b], Insa-Franco R.[b].

[a] Planning Department Office, Government of Antioquia, Medellín, Colombia
[b] Department of Transport Engineering and Infrastructure, Universitat Politècnica de València (UPV), Valencia, Spain

Abstract

Railways are an efficient transport mean with lower energy consumption and emissions in comparison to other transport means for freight and passengers, and yet there is a growing need to increase their efficiency. To achieve this, it is needed to accurately predict their energy consumption, a task which is traditionally carried out using deterministic models which rely on data measured through money- and time-consuming methods. Using four basic (and cheap to measure) features (train speed, acceleration, track slope and radius of curvature) from MetroValencia (Spain), we predicted the traction power using different machine learning models, obtaining that a random forest model outperforms other approaches in such task. The results show the possibility of using basic features to predict the traction power in a metropolitan railway line, and the chance of using this model as a tool to assess different strategies in order to increase the energy efficiency in these lines.

Keywords: Machine Learning; Traction power; Random Forests; Metropolitan railway lines; Energy consumption; Artificial Neural Networks

1. Introduction

In current times of climatic change and increasing global energy consumption and emissions, there is a particular interest in improving energy efficiency in the transport sector, which is important for all human activities.

According to the International Energy Agency [1], in 2016 there was an overall energy consumption of 9,555 MTOE (millions of tonnes oil equivalent) of which 28.8% correspond to energy consumed by the transport sector, hence demonstrating its great contribution to worldwide energy consumption. Railways are, compared to other transport means, rather efficient, as they, for instance, only represent about 2% of the energy consumed by the transportation sector in the EU-28, despite carrying up to 17% of the freight and 8% of the passengers across the European Union [2]. On the other hand, the United Nations [3] has estimated that 68% of the world's population will live in urban areas in the year 2050, so that any strategy carried out in cities for reducing energy consumption will have a greater impact on addressing this issue on a global scale.

Following this thread, metropolitan railway lines have proven to be more efficient, by far, than other types of vehicles for passenger transport in cities [4]. However, as these systems tend to involve large-scale operations and aim to offer high frequencies, they consume a large amount of energy. Moreover, energy consumption may represent up to 25% of the total life cycle cost of

a passenger train, and thus it has direct impact on its sustainability [5]. In addition, the energy consumption in a metropolitan railway line is mainly due to the traction, followed by air conditioning, lighting and drainage, and other auxiliary systems. Hence, making any reduction in traction energy has great potential to improve the energy efficiency and competitiveness of metro systems [4]. Therefore, many strategies have been implemented with the aim of reducing the energy consumption in railways, which focus on operations, rolling stock and track geometry optimisation. Strategies related to operations include using driving simulators to improve manually driven trains [6,7] or to equip trains with automatic train operation [8]. Alternatively, some authors have proposed to modify rolling stock characteristics such as the implementation of regenerative brake [9], on board storage systems [10] and considering train load variations and delays [11]. Moreover, other authors have focused on optimising track geometry [12–15].

Modelling and prediction of energy consumed by metropolitan railway lines has been addressed by different authors using different approaches. Deterministic models based on the Davis equation are the most common method for modelling the energy consumption on trains because they generally yield a good enough approximation of the energy consumed and it is relatively easy to increase the complexity of the model [16]. Other authors have implemented this approach adding features such as unexpected delays [17], regenerative brake [18], variations of train mass [19,20], on-board storage systems [21], or consumption of auxiliary systems [22]. The vast majority of those features require specific monitoring devices, to gather data essential to validate such models, and thus obtaining solutions may be quite money- and time-consuming. However, it is necessary to measure features regarding the train dynamics and basic track layout to implement deterministic models, which may increase their cost of implementation [23].

One alternative to deterministic models are machine learning (ML) approaches, which include several techniques that allow computers to mechanise data-driven model programming and build models by means of a methodical detection of non-linear connections between data [24–27]. Additionally, De Martinis and Corman [28] presented the potential of using data-driven approaches for improving energy efficiency in railways.
The most common ML model used for predicting the energy consumption in metropolitan railway lines is the artificial neural network model (ANN), which has even been combined with deterministic models.

Chuang et al. [29] used an ANN model to determine the optimal coasting speed of train operation for the Kaohsiung mass rapid transit system to achieve the cost minimisation of energy consumption and passenger traveling time.

Komyakov et al. [30] performed the analysis of the main factors influencing the consumption of electricity for rail transport and developed an ANN model to simulate the power consumption of a complete railway network, instead of a single train. On the other hand, Chen et al. [31] established ANN models to calculate the position of multiple trains. Similarly, Pineda-Jaramillo et al. [14] developed an ANN using consumption data measured in MetroValencia to estimate the energy consumption of the train, later used the ANN for testing hypothetical operational scenarios aimed to reduce the energy consumption of a metro system, including different vertical alignments, and then analysed the impact of the construction costs of these vertical alignments.

Huang et al. [32] built an ANN model combined with a deterministic model to describe the nonlinear characteristics of train braking. In the same way, Dündar and Sahin [33] developed an

ANN combined with deterministic models to mimic the behaviour of train operators, and tested it with data extracted from conflict resolutions in train operations in Turkish State Railways.

Finally, it is important to note that studies using ML approaches other than ANN for predicting railways energy consumption are rather scarce. In fact, we only found the work of Yu et al. [34] who estimated the daily electricity consumption of a newly constructed metro station, rather than on a train itself, using a Support Vector Machine model.

As the literature reviewed show, the energy consumption in railway lines has often been estimated using deterministic models which require specific monitoring devices to gather the data required for model validation. This solution is money- and time-consuming.

On the other hand, ML models, and specifically ANN models, have been tested as an alternative and have demonstrated a reasonable degree of accuracy according to the reviewed studies, but in many cases, researchers implement this model blindly, disregarding some of its inadequacies such as its difficulties to present interpretable results (for this reason, it is common for many researchers to refer to ANN models as “black-box models”) [35]. In addition, there is a lack of studies analysing the potential of other simpler ML models with more interpretability advantages, to predict the energy consumed in metropolitan railway lines using basic measurable features such as train speed, acceleration and track geometry characteristics.

Within this framework, this paper aims to assess the potential of a range of ML models to be used as tools to predict traction power in metropolitan railway lines using four features as predictors: train speed, acceleration, track slope and radius of curvature. In order to do so, we will implement six ML models (linear regression, ridge regression, decision trees, random forests, gradient boosting and artificial neural network) and we will compare their performance in achieving this particular task. These models were chosen due its popularity for solving regression problems, and due the configuration of the dataset, where we have an intermediate number of observations (approximately 50,000) and a small number of predictors; moreover, these methods have either been successfully used in transport research or have shown promising results in many fields [27,36].

This paper is organised in four sections. Section 2 describes the materials and methods used for predicting the energy consumption in metro trains using ML approaches, including a background of the ML models used in this research. Section 3 presents the results and discussion. Finally, section 4 presents the conclusions of the study.

2. Materials and methods

2.1. Dataset

As explained before, reliable data is required to validate any model. In this case, as we aim to predict the traction power of a metropolitan railway line, three MSAVDC meter devices (by Mors-Smitt ®) were installed in a metro unit to measure the consumed (and generated) power in the pantograph as well as the power consumed by auxiliary systems and dissipated in the rheostatic brake. The data was measured in real time (with a sampling frequency of 1 Hz), in lines 1, 2, 3 and 5 of MetroValencia (Spain). This metro network operates at 1500 V DC, and the trains are equipped with regenerative brake. However, they lack any kind of energy storage system, and thus regenerated energy is either consumed by the train itself (for auxiliary systems), returned to the catenary (but only if there is another train accelerating nearby), or dissipated in the rheostatic brake.

Train speed was measured using a Knorr sensor model BB0457681100. For more detail on the monitoring methodology, see our previous study [26]. In total, we collected a dataset of 52,322 records.
Data was complemented with the track slope and radius of curvature of such lines, provided by the infrastructure manager (resolution: 1 mm); however, this data on track layout was incomplete with regards to lines 1 and 2, and thus was available only for 58.3% of the dataset (30,529 records). For these reasons, the data analysis will be presented for the dataset consisting on 30,529 observations and the predictors presented in Table 1, but the models will be considering both cases (i.e. partial dataset with track layout and full dataset) in section 3. It is worth noting that, for the purposes of this research, the train is considered as a point mass, and thus the values of track slope and curvature used at every step of the calculations explained below are those located right below the train's mass centre.

Table 1. Attributes contained in the dataset

Attribute	**units**
Train speed	km/h
Acceleration	m/s^2
Track slope	mm/m
Radius of curvature	m
Consumed power in the pantograph	W
Generated power in the pantograph	W
Consumed power in the rheostatic brake	W
Consumed power in the auxiliary systems	W

2.2. Pre-processing

In order to make the data ready for deeper analysis, we carried out a data wrangling of the full dataset, performing different tasks such as removing rows with no full information (or filling them with median/mean values), merging available data, etc. We intend to model total traction power (and thus that is our target feature) and not energy, because that is what our monitoring devices provide. That said, power and energy are directly related, and the former informs the latter. Moreover, as we seek to model the power exclusively used for traction, the initial four variables related to power where combined according to Equation (1).

$$Traction_{power} = Consumed\_power_{pantogaph} + Generated\_power_{pantograph} - 2 * Consumed\_power_{auxiliaries} - 2 * Consumed\_power_{resistances} \quad \text{Equation (1)}$$

As Equation 1 shows, power consumed in auxiliary systems and dissipated in the rheostatic brake were multiplied by 2. The reason to do so is that the monitored train is a multiple unit with two rheostatic brakes and two distribution panels for auxiliary systems, but only one of each was equipped with a monitoring device. However, according to the information provided by the operation manager, it is safe to assume that each auxiliary panel supplies half of total needs of the train, and that the regenerated energy is dissipated roughly equally in both resistances.

In addition, the maximum power is limited by the total train power, including the IGBT converters, plus the efficiency of the electrical motors and the energy loses between the pantograph and the traction motors. Assuming and overall efficiency of 70% in the traction chain, this yields an upper value around 2,100 kW. Data from energy meters show maximum values of 2,055 kW, meaning for the 98 % of the upper value. Therefore, the major part of the power range is covered by the measurements and thus taken into account in these analyses.

On the other hand, due the nature of the data which comprises larger scales, we did not remove outliers, but we scaled the data as we will explain later. Also, we observed that there was no collinearity between the predictors (see Table 2.), which is a good signal because a collinearity between predictors could cause problems such as redundancy (where two predictors provide the same information) so that the estimation done by these predictors on the target variable would be less reliable and less precise.
After the initial data wrangling, we obtained a clear dataset of 30,529 rows and 5 columns representing 5 features as presented in Table 3, providing a good sample size for predicting the traction power using ML approaches.

Table 2. Collinearity between the predictors

speed				
acceleration	0.03336			
slope	0.01005	0.00741		
curvh	0.10114	0.00312	0.02122	
	speed	**acceleration**	**slope**	**curvh**

Table 3. Description of the 5 features of the cleaned dataset

	Mean	Standard deviation	Min Value	Q1: 25%	Median: 50%	Q3: 75%	Max value
speed	27.4	22.6	0.0	0.0	27.2	45.8	77.6
acceleration	0.0	0.4	-1.3	-0.2	0.0	0.0	1.1
slope	-0.1	9.5	-26.0	-1.9	0.0	1.7	26.0
curvh	146.7	427.5	0.0	0.0	0.0	0.0	4760.2
traction_power	150.5	539.5	-1615.4	-50.7	9.4	142.2	2021.9

To perform the training of the ML models, we split the data into a set of all the input features (train speed, acceleration, track slope and radius of curvature, the latter denoted as curvh) as '*X'* and the target feature (denoted as traction_power) as '*y'.* Then, we implemented the feature scaling method to the input features to scale the range of independent features. This method is used to improve the convergence of steepest descent algorithms and to avoid a situation when several features dominate other features in magnitude [37,38].

2.3. Training of the ML models

With the scaled input features and the target feature, we randomly split the dataset into a training subset (70% of the data) and a test subset (30% of the data). The first subset is used to train the models, whereas the last subset is used to validate them. It is important to check that the distribution of the values of both groups remain similar, as shown in Figure 1.

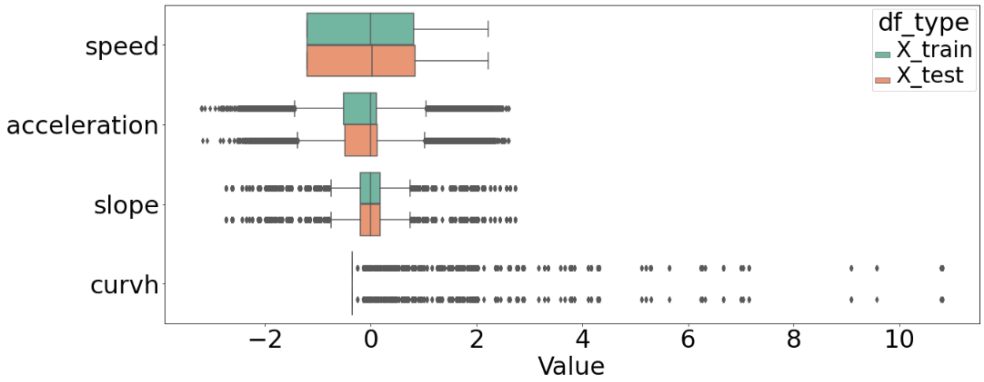

Figure 1. Similarities in value distribution for each feature in training and test data

Considering that the target feature is traction power, which is a numeric feature, the most suitable ML approach is regression. Fundamentally, regression models are supervised ML models used for predicting numeric values.
A supervised ML model, supplied with a series of *input* (independent) features (such as speed, acceleration, and slope) which will work as predictors, and a *target* (dependent) feature to predict (e.g. traction power). Then the *output* provided by the ML model is compared with the *target* feature, and the model parameters are adjusted through an iterative process until a good agreement between the model and reality is achieved [26].
Therefore, to predict the traction power in a metropolitan railway line, we trained different ML models using more than 30,000 records. Analyses were done using Python programming language, version 3.6. Below we present a short explanation of the regressor models used in this study, which have been vastly documented in several studies.

2.3.1. Linear and ridge regressions

Linear regression (LR) is often used to estimate a model for predicting future responses, or to research the relationship between predictor features and the *target* feature. For the former, model prediction accuracy is important, while for the latter the size of the model is of more interest [39,40].

Ridge regression and the lasso are regularised versions of least squares regression using penalties on the coefficient vector, which is commonly used for estimation of generalized linear models with convex penalties, and the so-called *elastic net* which is a mixture of the ridge and lasso methods [41]. These methods are used to overcome some flaws of the linear regression, such as not performing well with respect to both prediction accuracy and model size [42].

2.3.2. Tree-based ensemble techniques: decision trees (DT), random forests (RF) and bagging classifiers (BC)

A decision tree (DT) model is a ML model which is generally preferred over linear regression models when the data has many features which interact in nonlinear ways. Linear regression models use a general predictive formula holding the full dataspace while decision trees split the space into smaller regions and further partition the subdivisions and assigns to its nodes where interactions are more controllable. Regression trees regress decisions in a tree form, starting from the root node down to a leaf node where the leaf node contains the response [43]. A DT model has three sensitive hyperparameters which need to be calibrated for good performance: the number of trees of the model, the minimum number of samples to split an internal node, and maximum depth of the tree.

A tree-based ensemble method is the random forest (RF) model, which generates many regressors and aggregates their results. They are particular useful for solving regression, classification, and other tasks. The RF technique uses several decision trees (as regressors) in parallel for parametrization, and aggregates their results by integrating a sampling technique and an ensemble approach to optimise model building [44]. RFs can be seen as a predictor tool which consists in a collection of tree-structured regressors with independent identically distributed random vectors [45]. A RF model has three sensitive hyperparameters which need to be calibrated for good performance: number of trees, the minimum number of samples to split an internal node, and maximum depth. Moreover, it is necessary to define the minimum number of samples required to be at a leaf node to prevent overfitting.

Another thee-based ensemble method is the gradient boosting (GB) model, which works through a sequential process where each successive model tries to fix the errors of the preceding model. Hence, every successive model depends on the preceding model. Each single model might not achieve a good accuracy for the entire dataset, but they will perform well for some fragment of the dataset. Therefore, each of the single models substantially improves (boosts) the performance of the ensemble. Hence, the boosting techniques combine together several weak-learners to form a strong learner [46]. A GB model has three sensitive hyperparameters which need to be calibrated for good performance: number of gradient boosted trees, the boosting learning rate, and maximum tree depth for base learners.

2.3.3. Artificial neural networks (multi-layer perceptron)

Artificial neural networks (ANN) extract complex patterns from within the data, and detect complex trends by deriving meaning from complex or inaccurate data [47,48]. An ANN model is formed by *neurons*, which are elements that are interconnected and work in unison to solve diverse problems. The ANN is trained using *input* and *target* data (in the same way all supervised ML models work), where the available *target* data is linked with *output* data provided by the ANN, then the ANN parameters are adjusted using an iterative process until an optimal adjust among the true data (*target*) and the model (*output*) is achieved [26].

One popular type of ANN model is the multi-layer perceptron (MLP), which sets a function $f(.):RD \rightarrow Ro$, by training on a dataset, where '*D'* means the dimension of the *input* dataset, and '*o*' means the number of dimensions of expected output. Given '*X'* set of features, and a target '*y'*, where $X=\{x1,x2,x3,\ldots,xD\}$, this model can learn a non-linear function approximator for regression and classification tasks [46]. An MLP model has four sensitive hyperparameters which need to be calibrated for good performance: the activation function for the hidden layer, the learning rate, the number of neurons in the hidden layer, and the solver for the weight optimisation.

2.4. Validation of the models

We used the 10-fold stratified cross-validation method to validate the ML models, which is the most suitable method to validate a regressor model of a sample data [49]. This method randomly splits the training dataset into ten sets. In each set, the class is characterised in roughly the same amounts as the entire dataset. Later, each section is seized out in turn, and the ML model is applied to the remaining nine sets. Consecutively, the evaluation metric is calculated on the holdout set. This process will diminish any bias that can be produced by the holdout method which assess a given fraction of the data to validate and uses the remnant for training.

2.5. Evaluation metrics

Typically, the most common metrics used to assess the performance of a regression model are the relative mean squared error and the R-squared coefficient.

The relative mean squared error (rMSE) is expressed according to Equation (2).

$$rMSE(\%) = \frac{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(Var(y_i))} \quad \text{Equation (2)}$$

Where y_i is the true value of the observation i (target), $\hat{y}_i$ is the predicted value of the observation i (output of the model) and *n* is the number of observations. This is a useful metric as it provides insight into the proportion of the target data variance not explained by the model. This value is applied comparatively rather than absolutely; in general, the lower value of rMSE, the better. Thresholds of accepted values of rMSE are arbitraries, but some authors have considered rMSE values as acceptable when they are lower than 25%, or, in other words, that only 25% or less of the data variance is not explained by the model [50,51].

Another very popular choice for evaluating regression is the *R-squared* coefficient. Its success is mainly due to the fact that it is easy to use and interpret. The reason for these advantages is that for classical linear models, least squares and maximum likelihood estimators coincide, so they are consistent estimators with a direct geometric interpretation [52]. In general, the higher value of *R-squared* coefficient, the better.

Overall, it is common to choose the model with a lesser value of rMSE and a greater value of R-squared coefficient, for implementing a deeper analysis to that model with the aim of optimising such values.

2.6. Tuning of the parameters of the chosen model

In order to get a machine learning model that works well in practice, there are a number of parameters that need to be tuned when training the model, and the best parameter setting is different for different datasets. In other words, it is possible to improve the results of a ML model which has already yielded good results through “hyper-parameter tuning”.
The two most common methods to perform this task are *random search* and *grid search*. They both perform a search based on a defined subset of the parameter space, where the parameters are specified using minimal and maximal values, and a number of steps for each parameter. Both methods optimise the ML parameters using the 10-fold stratified cross validation method and assess each model by applying the specified evaluation metrics. Moreover, it has been demonstrated that the *random search* method outperforms the *grid search* method over the same domain, because the first one is able to find models that are as good or better within a small fraction of computation time [53]. For this reason, we use the *random search* method for tuning the parameters of the chosen model.

2.7. Analysis of possible underfitting/overfitting problems

After obtaining the best tuned model, it is important to analyse if it presents underfitting/overfitting. In a first instance, we used the 10-fold stratified cross-validation method to validate the ML models as we stated previously, this is complemented with the *early-stopping* method to prevent the ML models from overfitting [54].
Later, we implemented two strategies to analyse whether the selected model presents underfitting/overfitting: first, we implemented the *learning curve* method [55,56], a sampling

method which monitors the increasing costs and performance of the model as larger amounts of data are used for training, and finds out when future costs outweigh future benefits; then, we implemented the *grid search* method to verify that the parameters obtained in the previous step do not overfit the tuned model. Also, this method can work to simplify the configuration of the model, trying to reduce the value of the parameters to reduce the complexity of the chosen model.

2.8. Application of variance-based sensitivity analysis method to the chosen ML model

Sensitivity analysis is the evaluation of how uncertainty in a model's output can be distributed to uncertainty in the model's input features [57]. Overall, sensitivity analysis methods allow to explore the model behaviour across a broad range of conditions. In order to perform a sensitivity analysis in this study, we used a factor-fixing approach, which allows to identify which feature, among all model' input features, can be fixed (i.e. assigned any value within its range) without any appreciable impact on the output of the model [58].

In this study, the Sobol's variance-based sensitivity method [59] is applied to the chosen ML model. This method has independence in the estimation of sensitivity indices (i.e. potential non-additivity in the model does not affect its sensitivity index), and it allows for the quantification of interaction effects in model behaviour.

After achieving the best tuned ML model, the model is applied to a random sample where the ranges of each feature are previously defined. Then, we implemented the Sobol's method in order to calculate the total-order indices for all the input features in the model, using the open-source Python library: SALib. The total-order index of the feature *i* measures the contribution of the output variance caused by a model input, including the contribution to the output variance by the single input feature *i* alone, and the contribution to the input variance caused by the interaction of the input feature *I* with the other input features [60]. Equation (3) presents the formulation of the total-order index.

$$S_{Ti} = 1 - \frac{V[E(Y|X_{\sim \mathrm{i}})]}{V(Y)} \qquad \text{Equation (3)}$$

Where V [E (Y | $X_{\sim i}$)] is the conditional variance of the expected value of Y fixing all factors except X_i, averaged over all values of X_i, and V(Y) represents the unconditional variance of the model output. For a better comprehension of the key equations used in Sobol's method, readers can refer to previous studies [57,61].

3. Results and discussion

As explained before we used train speed, acceleration, track slope and radius of curvature (and a mixture of them) as predictors using 30,529 records. In addition, we used train speed and acceleration as predictors using 52,322 records (as we mentioned in section 2.1, we do not have the information related to track slope and radius of curvature for parts of lines 1 and 2 that represent 41.7% of the data).

We trained the six ML models mentioned in section 2.3 using the scaled training data, then we assessed the models using the performance metrics detailed in section 2.5 on the test data. The results presented in Table 4 correspond to the mean of 10-fold after applying the cross-validation method.

According to these results, the RF model was the best model in terms of R-squared coefficient and rMSE applied to the test data for all the combinations of predictors and 30,529 number of

records. GB, DT and ANN offer results only slightly below those of the RF in terms of performance. On the other hand, LR and ridge models do not work well for predicting the traction power of metropolitan railway lines.
Additionally, we can see that the best results are obtained using the four predictors. Also, it is important to highlight the fact that if we use a larger number of records, there is an improvement in the model using only the train speed and the acceleration as predictors.
Considering these initial results, we performed a deeper analysis to the random forest model using the 30,529 records in which we have data for the four predictors.

Table 4. Results of the preliminary ML models

Number of records	Predictors used	Model	R2_test	rMSE_test [%]
30,529	Speed Acceleration Slope Curvature	Linear regression (LR)	-0.01	50.28
		Ridge regression (Ridge)	-0.01	50.28
		Decision tree (DT)	0.68	25.70
		Random forest (RF)	**0.70**	**22.69**
		Gradient boosting (GB)	0.66	24.76
		Neural network (ANN)	0.65	26.02
	Speed Acceleration Slope	Linear regression (LR)	-0.01	50.27
		Ridge regression (Ridge)	-0.01	50.27
		Decision tree (DT)	0.68	25.28
		Random forest (RF)	0.70	22.78
		Gradient boosting (GB)	0.66	24.77
		Neural network (ANN)	0.64	26.09
	Speed Acceleration	Linear regression (LR)	-0.11	52.69
		Ridge regression (Ridge)	-0.11	52.69
		Decision tree (DT)	0.62	29.28
		Random forest (RF)	0.64	27.00
		Gradient boosting (GB)	0.61	27.54
		Neural network (ANN)	0.59	28.97
52,322	Speed Acceleration	Linear regression (LR)	-0.05	50.95
		Ridge regression (Ridge)	-0.05	50.95
		Decision tree (DT)	0.64	27.77
		Random forest (RF)	0.66	25.46
		Gradient boosting (GB)	0.65	25.71
		Neural network (ANN)	0.64	26.36

In order to improve the RF model to work well in practice, we implemented the hyper-parameter tuning by applying the random search method to the training data (using the 10-fold stratified cross-validation method) to achieve the best combination of parameters of the RF model, obtaining a small improvement on the evaluation metrics.
The tuned model has the following hyperparameters: minimum number of samples to split an internal node:5, and number of trees:97 (Table 5).

Table 5. Results of the hyperparameter tuning for the Random Forest model

Number of records	*Predictors used*	*Model*	*R2_test*	*rMSE_test [%]*
30,529	*Speed Acceleration Slope Curvature*	*Initial random forest (from Table 4)*	0.70	22.69
		First tuned random forest (after applying the hyper-parameter tuning)	0.70	22.65

Finally, to validate that the tuned model does not present underfitting/overfitting, and trying to reduce the complexity of this model, we used the learning curve and grid search methods, and the results are shown in Figure 2 (a, b, c). According to the learning curve presented in Figure 2 (a) there is a convergence trend between both curves (and the gap between them is small) as a major number of samples (x-axis) are used to train the model, pointing out a good fitting of the model and it validates the fact that it is possible to improve the performance of the model by adding more records. This also can be validated by the fact that there was an improvement in the model using only the train speed and the acceleration as predictors (see Table 4).

On the other hand, in Figure 2 (b) we see that the rMSE (%) decreases as the number of trees (n_estimators axis) grows, but a change in the minimum number of samples required to split an internal node (min_samples_split axis) does not affect the performance of the model in terms of reduction of the rMSE.

Lastly, in Figure 2 (c) we see that in the range from 1 to 100 trees (n_estimators axis), there is a constant improvement of the model in terms of reduction of the rMSE for the training and test data, but from 25 trees onward, the rMSE of the test data tends to slowly increase.

According to Figure 2 (b and c), with the aim of having a simpler model with good accuracy and preventing overfitting/underfitting, we choose a minimum number of samples required to split an internal node (min_samples_split) of 2, and 25 trees (n_estimators), obtaining another small improvement on the evaluation metrics (see Table 6).

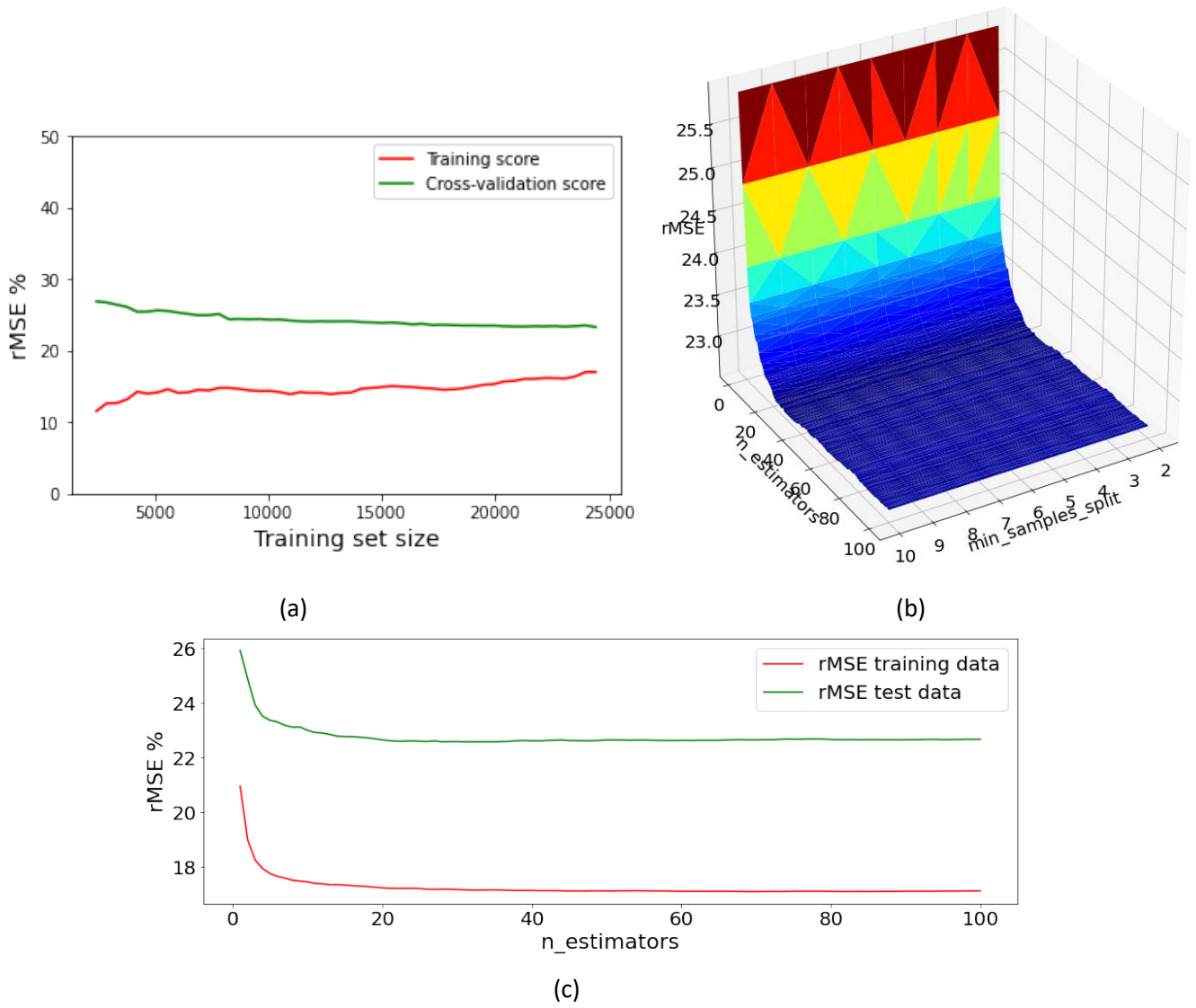

Figure 2. Analysing possible underfitting/overfitting problems in the RF model, by applying: (a) learning curve method for assessing the number of observations used for training the data, (b) grid search method for assessing the rMSE in the test data modifying the number of trees and the minimum number of samples required to split an internal node, and (c) grid search method for assessing the rMSE in the training data and in the test data modifying the number of trees (the minimum number of samples required to split an internal node was not used, due the previous verification in (b)).

Table 6. Results of the Random Forest model after second validation of the parameters

Number of records	Predictors used	Model	R2_test	rMSE_test [%]
30,529	Speed Acceleration Slope Curvature	Initial random forest (from Table 4)	0.70	22.69
		First tuned random forest (after applying the hyper-parameter tuning)	0.70	22.65
		Second tuned random forest (after a deeper analysis)	0.70	22.58

With the final model, we estimated the errors between the predicted traction_power and the measured traction_power in the test data, obtaining the distribution presented in Figure 3. As we can see, the distribution of the errors is quite gaussian, which can be considered as a good fitting of the model.

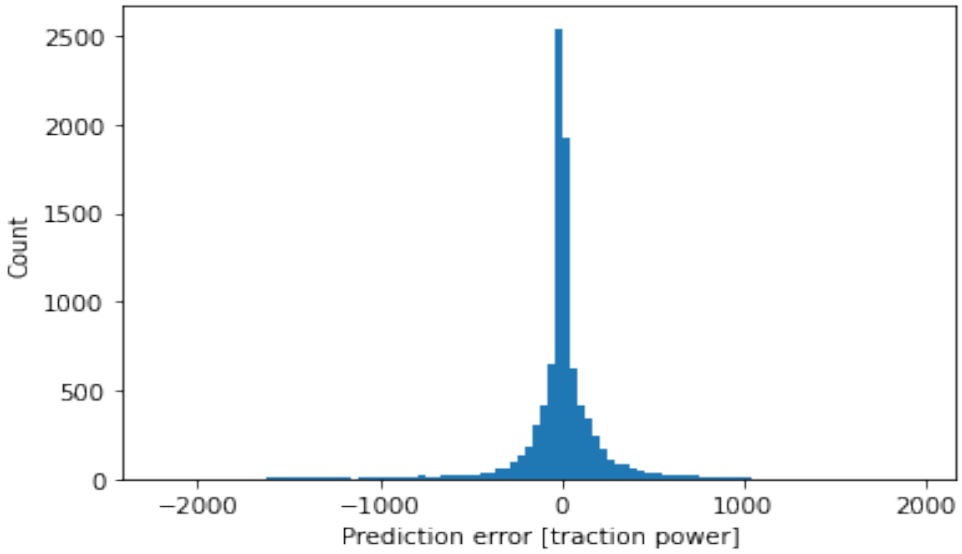

Figure 3. Distribution of errors between the predicted traction_power and the real traction_power in the test data

On the other hand, the resulting model gives us the importance of each of the four predictors used: acceleration: 65.7%, train speed: 29.1%, track slope: 4.7%, radius of curvature: 0.5%.

With the trained model, we generated a random sample of the four input features where the ranges of each feature are defined by the ranges of the original dataset (see Table 3). Then the model was applied to the random sample for computing the power traction. Finally, the SILab Python library was applied to the resulting values in order to estimate the total-order indices for all the input features in the model, obtaining the results presented in Table 7.

Table 7. Total effect indices of the input features of the model

Feature i	S_{Ti} estimate
speed	0.733
acceleration	1.065
slope	0.623
curvh	0.671

It is interesting to highlight that the two most significative features considered by the model to predict traction power are train acceleration and speed, which have been used by researchers to optimise the energy consumption in different railway lines. On the other hand, Table 7 shows that acceleration is the most-influential feature in the output of the RF model, and, although the slope and curvh are the less-influential features, they have an important weight in the estimation of the traction power. The total-order index can be used to identify a possible non-influential feature since the estimate includes the interaction effects between features, which is not computed by the importance features given by the RF model.

From these results, it is important to highlight that although apparently the track geometry does not influence in the RF model in order to compute the traction power, the interaction between these two features with speed and acceleration becomes important in such task.

Speed and acceleration have been validated for MetroValencia itself in previous studies [26,54,62]. These results validate the veracity of our model and the possibility of applying different ML approaches to obtain better results than the most common ML model used: artificial neural networks (see Table 4).

The key strength and most important implication of our results is the possibility of training a low-cost ML model with basic (and cheap to measure) features to predict the traction power in metropolitan railway lines. This is an important outcome considering that the required data to compute the traction power in a metropolitan railway line requires specific monitoring devices, a solution that is quite money- and time-consuming. These models still require speed profiles as input data. These profiles may be provided by deterministic models when testing new driving scenarios or may be the result of running heuristic optimisation models, which yield different optimised speed profiles fulfilling the respective restrictions. In any case, they provide a useful alternative to estimate energy consumption when engine performance and other train related data is not available or reliable. Other strength is that the model can be improved by adding more collected data as the learning curve shows (Figure 2 (a)). This model can also work as a tool to test strategies aiming to reduce the energy consumption in metropolitan railway lines.

4. Conclusions

This paper presents an assessment of the performance of six different ML models used to predict the traction power of a metropolitan railway line. This research was conducted to deal with the costs and complexity associated to the collection of data required to compute energy consumption using traditional methodologies (i.e. deterministic models). ML models are a valid alternative to such traditional methodologies, as they are powerful techniques that allow computers to mechanise data-driven model programming and build models by means of a methodical detection of non-linear connections between the data. On the other hand, we wanted to analyse different ML models other than artificial neural networks (which has been the most studied ML so far to model railways energy consumption), in order to test whether other ML models have better performance than ANN for achieving such task.

Among the models tested, the model that performs better in predicting the traction power of a metropolitan railway line is the random forest model. This model achieved an rMSE of 22.58% and a R-squared coefficient of 0.70. Furthermore, according to the learning curve, it is possible to improve this model by adding more collected data. Moreover, among the four features used as predictors to estimate the traction power of the train, we got that the acceleration is the most important feature in achieving such task, whereas the train speed, the track slope and the radius of the curvature are also important only if these features are used jointly.

A first contribution of this study is that it is possible to use a low-cost ML model trained with basic (and cheap to measure) features to predict the traction power in metropolitan railway lines, saving time and cost related to data collection in comparison to deterministic models.
Also, we demonstrated in this study that at least three ML models have better performance in predicting the traction power of a metropolitan railway line than the multi-layer perceptron – artificial neural network using the train speed, acceleration, track slope and radius of curvature as predictors. Therefore, it is possible to use these models as a tool for developing different strategies to reduce the energy consumption in metropolitan railway lines. This would be a next step within a larger research scheme, where these models would be applied to test different scenarios (considering vehicle characteristics, changes in track layout, different schedules and other factors) and determine the more efficient ones in terms of energy consumption.

The main limitation of our research is the fact that we used only one metropolitan railway line to train the ML models, so it is not possible to prove transferability of our methodology and outcomes. However, this could be used in future research to test and improve our models using similar datasets from different railway lines, not only metropolitan railway lines, in order to generalise this methodology. Also, future researches can use this ML model as a tool for analysing different strategies (such as driving patterns or on-board storage systems) with the aim of reducing the energy consumption in railway lines.

5. Acknowledgements

This paper was possible thanks to the collaboration agreement signed between the *Universitat Politècnica de València* and *Ferrocarrils de la Generalitat Valenciana*, and funding obtained by the Spanish Ministry of Economy and Competitiveness, through the project *"Strategies for the design and energy-efficient operation of railway and tramway infrastructure"* (Ref. TRA2011-26602). In addition, we would like to express our gratitude to two anonymous reviewers for their valuable comments on an early version of this article.

6. References

[1] International Energy Agency. Key World Energy Statistics 2018. Paris; 2018.
[2] European Commission. EU transport in figures - Statistical Pocketbook. 2017. DOI:10.2832/041248
[3] United Nations. World Urbanization Prospects: The 2018 Revision. World Urban. Prospect. 2018 Revis. New York: UN; 2019. DOI:10.18356/b9e995fe-en
[4] Su S, Tang T, Wang Y. Evaluation of strategies to reducing traction energy consumption of metro systems using an Optimal Train Control Simulation model. Energies. 2016;9(2):1–19. DOI:10.3390/en9020105
[5] UIC. Process, Power, People - Energy Efficiency for Railway Managers. Paris; 2008.
[6] Sicre C, Cucala P, Fernández A, et al. A method to optimise train energy consumption combining manual energy efficient driving and scheduling. WIT Trans Built Environ. 2010.

p. 549–560. DOI:10.2495/CR100511
[7] Salvador-Zuriaga P, García Román C, Pineda-Jaramillo JD, et al. The use of driving simulators for enhancing train driver's performance in terms of energy consumption. Libr Actas CIT2016 XII Congr Ing del Transp. 2016;1–14. DOI:10.4995/cit2016.2016.4236
[8] Brenna M, Foiadelli F, Longo M. Application of Genetic Algorithms for Driverless Subway Train Energy Optimization. Int J Veh Technol. 2016;2016:1–14. DOI:10.1155/2016/8073523
[9] Bocharnikov YV, Tobias AM, Roberts C, et al. Optimal driving strategy for traction energy saving on DC suburban railways. IET Electr Power Appl. 2007;1(5):675. DOI:10.1049/iet-epa:20070005
[10] Domínguez M, Fernández A, Cucala AP, et al. Efficient design of Automatic Train Operation speed profiles with on board energy storage devices. WIT Trans Built Environ. 2010. p. 509–520. DOI:10.2495/CR100471
[11] Fernandez-Rodriguez A, Fernandez-Cardador A, Cucala AP, et al. Design of Robust and Energy-Efficient ATO Speed Profiles of Metropolitan Lines Considering Train Load Variations and Delays. IEEE Trans Intell Transp Syst. 2015;16(4):2061–2071. DOI:10.1109/TITS.2015.2391831
[12] Samanta S, Jha MK. Modeling a rail transit alignment considering different objectives. Transp Res Part A Policy Pract. 2011;45(1):31–45. DOI:10.1016/j.tra.2010.09.001
[13] Lai X, Schonfeld P. Optimizing rail transit alignment connecting several major stations. Transp Res Board. Washington, DC; 2010.
[14] Pineda-Jaramillo JD, Salvador-Zuriaga P, Martínez-Fernández P, et al. Impact of Symmetric Vertical Sinusoid Alignments on Infrastructure Construction Costs: Optimizing Energy Consumption in Metropolitan Railway Lines Using Artificial Neural Networks. Urban Rail Transit. 2020; DOI:10.1007/s40864-020-00130-7
[15] Pineda-Jaramillo JD, Salvador-Zuriaga P, Insa-Franco R. Comparing energy consumption for rail transit routes through Symmetric Vertical Sinusoid Alignments (SVSA), and applying artificial neural networks. A case study of Metro Valencia (Spain). Dyna. 2017;84(203):17–23. DOI:10.15446/dyna.v84n203.65267
[16] Martínez Fernández P, Villalba Sanchís I, Yepes V, et al. A review of modelling and optimisation methods applied to railways energy consumption. J Clean Prod. 2019;222:153–162. DOI:10.1016/j.jclepro.2019.03.037
[17] Cucala AP, Fernández A, Sicre C, et al. Fuzzy optimal schedule of high speed train operation to minimize energy consumption with uncertain delays and driver's behavioral response. Eng Appl Artif Intell. 2012;25(8):1548–1557. DOI:10.1016/j.engappai.2012.02.006
[18] Lu S, Wang MQ, Weston P, et al. Partial Train Speed Trajectory Optimization Using Mixed-Integer Linear Programming. IEEE Trans Intell Transp Syst. 2016;17(10):2911–2920. DOI:10.1109/TITS.2016.2535399
[19] Carvajal-Carreño W, Cucala AP, Fernández-Cardador A. Optimal design of energy-efficient ATO CBTC driving for metro lines based on NSGA-II with fuzzy parameters. Eng Appl Artif Intell. 2014;36:164–177. DOI:10.1016/j.engappai.2014.07.019
[20] Liu R, Li S, Yang L, et al. Energy-Efficient Subway Train Scheduling Design With Time-Dependent Demand Based on an Approximate Dynamic Programming Approach. IEEE Trans Syst Man, Cybern Syst. 2020;50(7):2475–2490. DOI:10.1109/TSMC.2018.2818263
[21] Dominguez M, Fernandez-Cardador A, Cucala AP, et al. Energy Savings in Metropolitan Railway Substations Through Regenerative Energy Recovery and Optimal Design of ATO Speed Profiles. IEEE Trans Autom Sci Eng. 2012;9(3):496–504. DOI:10.1109/TASE.2012.2201148
[22] Huang Y, Yang L, Tang T, et al. Joint train scheduling optimization with service quality

and energy efficiency in urban rail transit networks. Energy. 2017;138:1124–1147. DOI:10.1016/j.energy.2017.07.117
[23] Khmelnitsky E. On an optimal control problem of train operation. IEEE Trans Automat Contr. 2000;45(7):1257–1266. DOI:10.1109/9.867018
[24] Bhavsar P, Safro I, Bouaynaya N, et al. Machine Learning in Transportation Data Analytics. In: Chowdhury M, Apon A, Dey K, editors. Data Anal Intell Transp Syst. Amsterdam: Elsevier; 2017. p. 283–307. DOI:10.1016/B978-0-12-809715-1.00012-2
[25] Allah Bukhsh Z, Saeed A, Stipanovic I, et al. Predictive maintenance using tree-based classification techniques: A case of railway switches. Transp Res Part C Emerg Technol. 2019;101:35–54. DOI:10.1016/j.trc.2019.02.001
[26] Pineda-Jaramillo JD, Insa-Franco R, Martínez-Fernández P. Modeling the energy consumption of trains by applying neural networks. Proc Inst Mech Eng Part F J Rail Rapid Transit. 2018;232(3):816–823. DOI:10.1177/0954409717694522
[27] Pineda-Jaramillo JD. A review of Machine Learning (ML) algorithms used for modeling travel mode choice. Rev DYNA. 2019;86(211):32–41. DOI:10.15446/dyna.v86n211.79743
[28] De Martinis V, Corman F. Data-driven perspectives for energy efficient operations in railway systems: Current practices and future opportunities. Transp Res Part C Emerg Technol. 2018;95(August):679–697. DOI:10.1016/j.trc.2018.08.008
[29] Hui-Jen Chuang, Chao-Shun Chen, Chia-Hung Lin, et al. Design of Optimal Coasting Speed for MRT Systems Using ANN Models. IEEE Trans Ind Appl. 2009;45(6):2090–2097. DOI:10.1109/TIA.2009.2031898
[30] Komyakov AA, Erbes VV, Ivanchenko VI. Application of artificial neural networks for electric load forecasting on railway transport. 2015 IEEE 15th Int Conf Environ Electr Eng. Rome: IEEE; 2015. p. 43–46. DOI:10.1109/EEEIC.2015.7165296
[31] Chen D, Han X, Cheng R, et al. Position calculation models by neural computing and online learning methods for high-speed train. Neural Comput Appl. 2016;27(6):1617–1628. DOI:10.1007/s00521-015-1960-6
[32] Huang Y, Tan L, Chen L, et al. A neural network driving curve generation method for the heavy-haul train. Adv Mech Eng. 2016;8(5). DOI:10.1177/1687814016647883
[33] Dündar S, Şahin İ. Train re-scheduling with genetic algorithms and artificial neural networks for single-track railways. Transp Res Part C Emerg Technol. 2013;27:1–15. DOI:10.1016/j.trc.2012.11.001
[34] Yu Z, Bai Y, Fu Q, et al. An Estimation Model on Electricity Consumption of New Metro Stations. J Adv Transp. 2020;2020:1–11. DOI:10.1155/2020/3423659
[35] Wu ZF, Li J, Cai MY, et al. On membership of black-box or white-box of artificial neural network models. 2016 IEEE 11th Conf Ind Electron Appl. Hefei: IEEE; 2016. p. 1400–1404. DOI:10.1109/ICIEA.2016.7603804
[36] Hagenauer J, Helbich M. A comparative study of machine learning classifiers for modeling travel mode choice. Expert Syst Appl. 2017;78:273–282. DOI:10.1016/j.eswa.2017.01.057
[37] Bollegala D. Dynamic feature scaling for online learning of binary classifiers. Knowledge-Based Syst. 2017;129(1):97–105. DOI:10.1016/j.knosys.2017.05.010
[38] Shyamala Devi M, Mathew RM, Suguna R. Regressor fitting of feature importance for customer segment prediction with ensembling schemes using machine learning. Int J Eng Adv Technol. 2019;8(6):952–956. DOI:10.35940/ijeat.F8255.088619
[39] Ruppert D. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. J Am Stat Assoc. 2004; DOI:10.1198/jasa.2004.s339
[40] van der Kooij AJ. Prediction accuracy and stability of regression with optimal scaling transformations [Dissertation]. Leiden University, Leiden, Netherlands; 2007.
[41] Zou H, Hastie T. Regularization and variable selection via the elastic net. J R Stat Soc

Ser B (Statistical Methodol). 2005;67(2):301–320. DOI:10.1111/j.1467-9868.2005.00503.x
[42] Martínez-Martínez JM, Escandell-Montero P, Soria-Olivas E, et al. Regularized extreme learning machine for regression problems. Neurocomputing. 2011;74(17):3716–3721. DOI:10.1016/j.neucom.2011.06.013
[43] Yildiz B, Bilbao JI, Sproul AB. A review and analysis of regression and machine learning models on commercial building electricity load forecasting. Renew Sustain Energy Rev. 2017;73:1104–1122. DOI:10.1016/j.rser.2017.02.023
[44] Suthaharan S. Machine Learning Models and Algorithms for Big Data Classification. Integr. Ser. Inf. Syst. 2016. DOI:10.1007/978-1-4899-7641-3
[45] Breiman L. Random forests. Mach Learn. 2001;45(1):5–32. DOI:10.1023/A:1010933404324
[46] Nti IK, Adekoya AF, Weyori BA. A comprehensive evaluation of ensemble learning for stock-market prediction. J Big Data. 2020; DOI:10.1186/s40537-020-00299-5
[47] Karlik B. Machine Learning Algorithms for Characterization of EMG Signals. Int J Inf Electron Eng. 2014;4(3):189–194. DOI:10.7763/ijiee.2014.v4.433
[48] Jordan MI, Bishop CM. Neural Networks. Stat Model Manag. 1 Oliver's Yard, 55 City Road, London EC1Y 1SP United Kingdom: SAGE Publications Ltd; 2004. p. 153–176. DOI:10.4135/9781446220566.n6
[49] Witten IH, Frank E, Hall MA, et al. Data Mining: Practical Machine Learning Tools and Techniques. 4th Editio. Data Min. Pract. Mach. Learn. Tools Tech. Elsevier; 2016.
[50] Molines J, Herrera MP, Medina JR. Estimations of wave forces on crown walls based on wave overtopping rates. Coast Eng. 2018;132:50–62. DOI:10.1016/j.coastaleng.2017.11.004
[51] Molines J, Medina JR. Calibration of overtopping roughness factors for concrete armor units in non-breaking conditions using the CLASH database. Coast Eng. 2015;96:62–70. DOI:10.1016/j.coastaleng.2014.11.008
[52] Ricci L. Adjusted -squared type measure for exponential dispersion models. Stat Probab Lett. 2010;80(17–18):1365–1368. DOI:10.1016/j.spl.2010.04.019
[53] Bergstra J, Bengio Y. Random search for hyper-parameter optimization. J Mach Learn Res. 2012;13(10):281–305
[54] Martínez Fernández P, Salvador Zuriaga P, Villalba Sanchís I, et al. Neural networks for modelling the energy consumption of metro trains. Proc Inst Mech Eng Part F J Rail Rapid Transit. 2020;234(7):722–733. DOI:10.1177/0954409719861595
[55] Meek C, Thiesson B, Heckerman D. The Learning-Curve Sampling Method Applied to Model-Based Clustering. J Mach Learn Res. 2002; DOI:10.1162/153244302760200678
[56] Richter AN, Khoshgoftaar TM. Learning Curve Estimation with Large Imbalanced Datasets. 18th IEEE Int Conf Mach Learn Appl. Boca Raton, FL, USA: IEEE; 2019. p. 763–768. DOI:10.1109/ICMLA.2019.00135
[57] Saltelli A, Annoni P, Azzini I, et al. Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index. Comput Phys Commun. 2010;181(2):259–270. DOI:10.1016/j.cpc.2009.09.018
[58] Jadun P, Vimmerstedt LJ, Bush BW, et al. Application of a variance-based sensitivity analysis method to the Biomass Scenario Learning Model. Syst Dyn Rev. 2017;33(3–4):311–335. DOI:10.1002/sdr.1594
[59] Sobol IM. Sensitivity analysis for nonlinear mathematical models. Math Model Comput Exp. 1993;1:407–414.
[60] Herman J, Usher W. SALib: An open-source Python library for Sensitivity Analysis. J Open Source Softw. 2017;2(9):97. DOI:10.21105/joss.00097
[61] Saltelli A, Ratto M, Andres T, et al. Global Sensitivity Analysis. The Primer. Chichester, UK: John Wiley & Sons, Ltd; 2007. DOI:10.1002/9780470725184

[62] Pineda-Jaramillo JD. Optimisation model of energy consumption in trains using symmetric vertical sinusoid alignments and its impact on infrastructure construction costs [Dissertation]. Polytechnical University of Valencia, Valencia, Spain; 2017. DOI:10.4995/Thesis/10251/90546