Machine Learning Supported Next-Maintenance Prediction for Industrial Vehicles

Sachit Mishra
Politecnico di Torino
Turin, Italy
sachit.mishra@studenti.polito.it

Luca Vassio
Politecnico di Torino
Turin, Italy
luca.vassio@polito.it

Luca Cagliero
Politecnico di Torino
Turin, Italy
luca.cagliero@polito.it

Marco Mellia
Politecnico di Torino
Turin, Italy
marco.mellia@polito.it

Elena Baralis
Politecnico di Torino
Turin, Italy
elena.baralis@polito.it

Riccardo Loti
Tierra Spa
Turin, Italy
rloti@tierratelematics.com

Lucia Salvatori
Tierra Spa
Turin, Italy
lsalvatori@topcon.com

ABSTRACT

Industrial and construction vehicles require tight periodic maintenance operations. Their schedule depends on vehicle characteristics and usage. The latter can be accurately monitored through various on-board devices, enabling the application of Machine Learning techniques to analyze vehicle usage patterns and design predictive analytics. This paper presents a data-driven application to automatically schedule the periodic maintenance operations of industrial vehicles. It aims to predict, for each vehicle and date, the actual remaining days until the next maintenance is due. Our Machine Learning solution is designed to address the following challenges: (i) the non-stationarity of the per-vehicle utilization time series, which limits the effectiveness of classic scheduling policies, and (ii) the potential lack of historical data for those vehicles that have recently been added to the fleet, which hinders the learning of accurate predictors from past data. Preliminary results collected in a real industrial scenario demonstrate the effectiveness of the proposed solution on heterogeneous vehicles. The system we propose here is currently under deployment, enabling further tests and tunings.

1 INTRODUCTION

Fleets of industrial and construction vehicles are subject to periodic maintenance. These vehicles are of different models and types. Thus, they require to plan actions of different type and frequency. According to the current vehicles' workload, maintenance schedule often changes. For example, some vehicles could remain unused for a relatively long period of time, then be moved to a construction site, and keep working at full capacity for many days or weeks. The heterogeneity of industrial vehicle usage has indeed prompted the need for tracking their utilization and automating the schedule of maintenance operations [3].

In the context of industrial vehicle management, the advent of CAN bus technology has enabled the design of data-driven decision-making processes [5]. The CAN bus provides access to various signals describing the vehicle usage state (e.g., working time, oil pressure, temperature, engine speed). Domain experts can thus monitor vehicle state in order to understand which maintenance actions need to be performed. In recent years, the significance advances of Internet of Things (IoT) and Big Data analytics technologies have fostered the development of smart predictive maintenance solutions [15]. Specifically, the analysis of CAN bus data by means of data mining and machine learning techniques allow us to (i) predict the future vehicle usage by means of classification and regression techniques (e.g., [7, 10]), (ii) aggregate vehicles with similar characteristics using clustering techniques, (e.g., [1, 4]), and (iii) identify malfunctioning of specific vehicle components (e.g., [6, 15]).

Optimizing maintenance activities of fleet vehicles is a priority in several industrial processes [2]. In fact, technological applications related to activity planning and resource management are crucial for efficiently handling logistics [14]. This paper proposes a machine learning approach to support the smart planning of the fleet maintenance operations in a industrial context.

Data-driven vehicle maintenance planning has already been addressed using various optimization methods. For example, in [11] the authors have applied Genetic Algorithms to plan the maintenance of geographically distributed assets by considering routing constraints and travel time to reach the assets. The authors in [12] have presented a dynamic optimization method to plan maintenance of heavy vehicles by jointly scheduling maintenance operations and production activities, whereas in [8] the authors propose a data-driven simulation framework for planning snow removal projects considering weather and truck-related data acquired by real-time sensors. All the aforesaid strategies are possible if accurate predictions of next maintenance events are available.

Contribution. This work presents an application of regression techniques to predict for a given vehicle when the next maintenance will be due. Specifically, it predicts the number of days left to the next maintenance operation, based on the series of past daily utilization levels and on the current time of usage left to the next maintenance. The application faces three of the main issues related to fleet maintenance planning: (i) *Vehicle heterogeneity*: The industrial vehicles in a fleet are commonly rather heterogeneous, in terms of number, type, and frequency of the necessary maintenance operations. This makes the planning of these activities particularly challenging and time-consuming

Published in the Workshop Proceedings of the EDBT/ICDT 2020 Joint Conference (March 30-April 2, 2020, Copenhagen, Denmark) on CEUR-WS.org.

for fleet managers. (ii) *Non-stationarity of the utilization series*: Vehicle usage pattern levels are rather irregular, whereas maintenance actions rather frequent. Seasonal trends may depend on vehicle type, model, and context of use. (iii) *Lack of data for new vehicles*: When a vehicle is added to the fleet, usage data is typically not available. This hinders the learning of accurate machine learning models, which requires historical data to train reliable predictors.

To handle heterogeneous fleets, we train a separate regression model per vehicle. Each regressor analyzes the vehicle usage patterns and the current time to maintenance of a specific vehicle. We conduct the analysis on 24 industrial and construction vehicles of different models. To handle the non-stationarity of the analyzed series, we incorporate the historical usage levels in the predictive models and train both linear and non-linear models. Finally, to overcome the lack of data related to new vehicles, we combine the regressor outcomes achieved on similar vehicles. The presented application is complementary to existing optimization-based planning strategies, e.g., [8, 11], providing the fleet management system with specific hints on future vehicle usage states. The results, achieved in real industrial scenario, show substantial improvements achieved by applying non-linear regression models compared to classical statistics-based or linear models. In light of the achieved results, the data owner (collecting telematics data from real industrial vechicles) has decided to put the present application under deployment, thus enabling further tests, tunings, and extensions.

This paper is organized as follows. Sections 2 formalizes the problem, while Section 3 describes the dataset and its preparation phase. Section 4 presents the data-driven methodologies. Section 5 summarizes the main experimental results. Finally, Section 6 draws conclusions and summarizes the future research agenda.

2 PROBLEM STATEMENT

Given an arbitrary industrial or construction vehicle v, our goal is to predict when the next maintenance operation for v will be due. Let N^v be the number of days for which historical data about v usage is available and let T^v be the allowed usage times (in seconds) for v between two consecutive maintenance operations. The period from one maintenance operation to the next one will be hereafter denoted as a *cycle*. The count of the number of days left to the next maintenance of vehicle v varies day by day. Let $D^v(t)$ be the series of the aforesaid daily counts. Our aim is to predict $D^v(t)$, where t denotes the current day. The series used to drive the prediction are enumerated below. For each vehicle v:

- $U^v(t)$: series of the daily utilization of vehicle v.
- $C^v(t)$: series of the counts of the number of days already passed from the last maintenance operation.
- $L^v(t)$: series of the utilization times left to the next maintenance operation. On an arbitrary day t, it is computed as follows:

$$L^v(t) = T^v - \sum_{i=t-C^v(t)}^{t-1} U^v(i) \tag{1}$$

In the following, we define three categories of vehicles according to the amount of historical data that is currently available: (i) *Old*: If at least one maintenance cycle has already been completed since data acquisition has started. (ii) *Semi-new*: If the first maintenance cycle has not been completed yet, but data about at least half of the usage in one cycle ($\frac{T^v}{2}$) is already available. (iii) *New*: If the vehicle has been used for less than $\frac{T^v}{2}$ seconds since the beginning of the data acquisition phase.

2.1 Error computation

To effectively support fleet managers in planning periodic vehicle maintenance, our prediction system is tailored to a specific goal that we encode by considering specific error function definition. We define three errors for each vehicle v: the *daily error* $E^v(t)$, the *global error* E^v_{Global}, and the *Mean Residual Error* $E^v_{MRE}(\widetilde{D})$.

The daily error counts on each day t the gap between the predicted and actual values of the next day of maintenance:

$$E^v(t) = \left|D^v(t) - D^v_{Predict}(t)\right| \tag{2}$$

The global error is a mean of the daily errors over all the N^v samples related to the vehicle under analysis, i.e.,

$$E^v_{Global} = \frac{\sum_{t=1}^{N^v} E^v(t)}{N^v} \tag{3}$$

The global error combines the daily errors together, but it does not consider nor weight the time that is left for the next maintenance. In other words, an error of 1 day when we are close to the maintenance (e.g., $D^v(t) = 1$) is considered as equal as an error when we are far from the maintenance (e.g., $D^v(t) = 100$). In order to solve this issue, the mean residual error is considered. It is the mean of the daily errors over specific days. In particular we want to compute the average only for specific values of D^v, contained in a set $\widetilde{D}$. $\widetilde{D}$ consists of a selection of days that are closer to the maintenance operation, for each maintenance cycle.[1] E^v_{MRE} is computed as follows:

$$E^v_{MRE}(\widetilde{D}) = \frac{\sum_{i:D^v(i)\in\widetilde{D}} E^v(i)}{\left|i : D^v(i) \in \widetilde{D}\right|} \tag{4}$$

The idea behind E^v_{MRE} is that fleet managers are mainly interested in getting accurate predictions when the vehicles are towards the end of their maintenance cycle, i.e., when maintenance operations need to be scheduled soon. Therefore, our main objective is to minimize $E^v_{MRE}(\widetilde{D})$.

3 DATA PREPARATION

The application presented in this paper has been developed and tested on real vehicle data provided by Tierra S.p.A.[2], a company that provides IoT solutions for monitoring vehicles of multiple vendors. The dataset consists of historical usage of 24 heterogeneous vehicles acquired over a 4 year period (from January 2015 to September 2019). For each vehicle, we consider the information coming from the CAN bus. Onboard sensors and Machine Control Systems generate messages for CAN at a frequency of approximately 100 Hz. Each message is collected by a controller which processes it, periodically generates a summary report, and sends it to a cloud server [7].

To prepare vehicle data for the present study, the input CAN bus data goes through a series of steps: (i) Data Cleaning, (ii) Normalization, (iii) Aggregation, (iv) Enrichment and (v) Transformation. A more detailed description of each of the above-mentioned steps is given below. Data cleaning entails properly handling missing values and inconsistent values. Data normalization allows us to scale the values of the utilization times to a uniform

[1] We have considered the last 29 days per cycle, i.e., $\widetilde{D} = \{1, \ldots, 29\}$.
[2] https://www.tierratelematics.com/

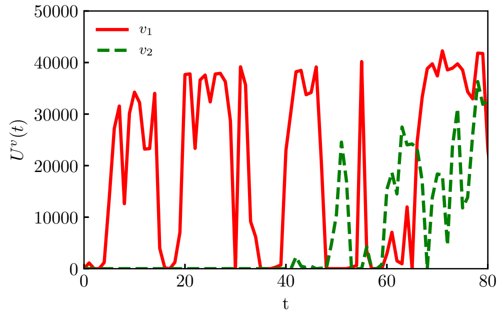

Figure 1: Daily utilization in seconds (U^v) in function of the day in the series (t) for two sample vehicles.

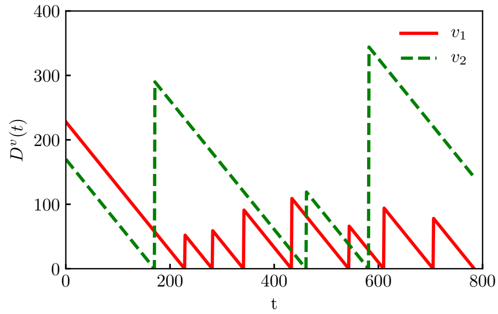

Figure 2: Target variable number of days left to the next maintenance (D^v) with respect to day in the series (t). Many cycles are shown for two sample vehicles.

value range (e.g., from 0 to 1) thus avoiding to introduce bias in regression model learning [13]. Data aggregation, enrichment and transformation aim at providing predictive algorithms with an appropriate set of features describing the usage patterns by aggregating data at the desired time granularity. For each vehicle we transformed the raw CAN bus data to produce the input features described in Section 2. Specifically, in our case of study, we primarily focus on daily-usage time series $U(t)$, i.e., the amount of time each vehicle worked on each day.

3.1 Preliminary data exploration

Figure 1 plots part of the series of daily utilization seconds, i.e., $U^v(t)$, for two sample vehicles. Curves show that vehicle utilization patterns are rather heterogeneous. Vehicle v_1 has a daily utilization of about 20 000-30 000 seconds, with few days without usage every 10-15 working days. On the other hand, vehicle v_2 remains almost unused for several weeks (from t=0 to t=40) and then suddenly changes its usage pattern.

After a fixed time amount of usage (we have considered T^v = 2 000 000 seconds), every vehicle needs to go under maintenance. Notice that we do not know whether maintenance operations have actually been performed or how long they take (from T^v on). Figure 2 shows two examples of target series $D^v(t)$, with many shown cycles. When $D^v(t)$ reduces to zero, the vehicle goes to maintenance. Then a new maintenance cycle starts, the number of days left to maintenance is maximal, and it monotonically decreases (one day for each day passed) until the next maintenance operation is carried out. Notice how v_1 has a first longer cycle (221 days), while the others are more constant and homogeneous, with length between 65 and 105 days.

In Figure 3 we show the number of days left to maintenance $D^v(t)$ with respect to the number of utilization seconds left for the next maintenance $L^v(t)$. The functions seems to have a constant rate when $L^v(t)$ is closer to zero, reflecting that for most of the time the utilization rate is relatively constant and different from zero. However, there are some vertical steps, corresponding to consecutive days when the utilization was null. This confirms that the presence of low- or zero-utilization periods has a serious impact on the target variable. Thus, predicting the correct target value could be challenging. Hopefully, it is unlikely to see long periods of zero-utilization in the days approaching the deadline. This reinforces the motivations behind using $E^v_{MRE}(\widetilde{D})$ as reference error metric, considering values relatively close to the maintenance.

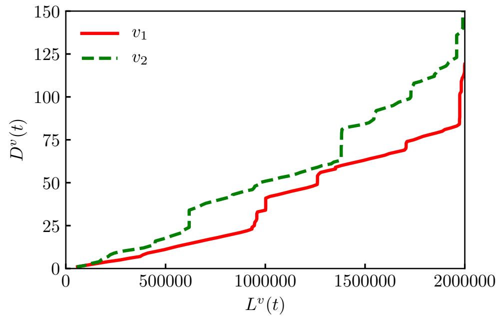

Figure 3: Utilization seconds left to maintenance (L^v) vs. the number of days to maintenance (D^v) for a single cycle of two vehicles.

4 METHODOLOGY

In this section we present the data mining process designed to address the problem under analysis. We propose different methodologies for vehicles when they are recognized as *new*, *semi-new*, or *old* (see Section 2). For each old vehicle we have enough data in order to train a prediction model on its own past. Semi-new and new vehicles will be separately handled.

For each vehicle v we generate a relational dataset containing the historical utilization series $U^v(x)$ [13]. More specifically, each record corresponds to a different day t and consists of a set of attributes denoting the past utilization levels (in seconds). Given a window size W, the attributes include the values $U^v(x)$ $[t - W \leq x \leq t - 1]$. Along with the utilization level series, the attributes include the current time left until the next maintenance, i.e., $L^v(t)$, and the target variable, i.e., the number of days left to maintenance, i.e., $D^v(t)$.

Many machine learning techniques work better with more data, hence we desire to increase the number of records per vehicle. Since we do not know when vehicle actually had the maintenance done, we can shift the time reference, i.e., changing the first starting day $t = 0$, without introducing errors. We randomly re-sampled multiple times the time reference starting from different time points within the training data and build the utilization series.

4.1 Approaches

We apply three different methods: (i) *a baseline* model, relying on simple estimate, (ii) *a univariate model*, whose predictions rely on a single variable, and (iii) *a multivariate model*, where the model considers multiple series values. These are standard methods for time series forecasting. Novelty relies in the categorization of the vehicles and data engineering.

4.1.1 Baseline algorithm: With this baseline approach we simply predict when the next maintenance will be due assuming that the utilization is constant and equal to the average utilization in the past. Hence, we compute the average utilization of vehicle v in the past (training set of size T_{train}). Then we use it for predicting the number of remaining days until the next maintenance is due. The average utilization is defined as follows:

$$AVG^v = \frac{\sum_{i=1}^{T_{train}} U^v(t)}{T_{train}} \tag{5}$$

Let $D^v_{BL}(t)$ be the number of days left to next maintenance predicted by the Baseline algorithm at time t as:

$$D^v_{BL}(t) = \frac{L^v(t)}{AVG^v} \tag{6}$$

We will denote the baseline method as BL thoroughout the paper.

4.1.2 Univariate regression model. We apply a univariate regression model $\mathcal{F}_{UR}$ to predict the number of days left to maintenance for a given vehicle v based on the last value of the daily utilization seconds series L^v:

$$D^v_{UR}(t) = \mathcal{F}_{UR}(L^v(t)) \tag{7}$$

4.1.3 Multivariate regression model. We extend the univariate regression model to a multivariate context in order to consider the temporal correlation between the target variable and the previous series values.

$$D^v_{MR}(t) = \mathcal{F}_{MR}(L^v(t), U^v(t-1), \ldots, U^v(t-W)) \tag{8}$$

Unlike the univariate model, the model formalized in Equation 8 does not consider only the last value of the daily utilization time but also the most recent values of the historical utilization series within a size-W window time interval, i.e., from $t-W$ to $t-1$ (where W is a user-specified parameter).

4.2 Regression algorithms

Univariate and multivariate regression models can be solved with linear or non-linear models. Linear models are deemed as appropriate whenever usage patterns are quite constant for most of the time. Conversely, non-linear models are potentially able to capture more complex, non-stationary usage trends. As a drawback, the complexity of non-linear models is typically higher than those of linear ones.

The deployed system allows fleet managers to select an acceptable trade-off between the accuracy and complexity based on empirical evidences. The results of a preliminary performance evaluation on real vehicle data are reported in Section 5. The models that have already been integrated and tested are briefly described below. A more detailed description is given in [13].

Linear Regression (LR): It is the simplest linear model. It learns a linear function minimizing the residual sum of squares between the predicted target value and the expected target value in the record of the training dataset.

Support Vector Regressor (SVR): It is among the most effective solutions to address regression and classification problems. Given a multidimensional training data representation, it finds an hyper-plane separating points belonging to different target value ranges. According to the kernel function used to derive the hyperplane, the predictive model can be either linear or non-linear (e.g., polynomial, sigmoid, rbf). Due to the high computational complexity of non-linear kernels, in the remaining of the paper we focus on linear SVR (LSVR).

Random Forest regressor (RF): It is an established ensemble method combining the predictions of multiple decision trees. Decision trees are the most popular non-linear mapping functions between non-predictive and predictive variables. They rely on tree-based structures. The Random Forest Regression averages the predictions made by various decision tree models, which are trained on different bootstraps (i.e., samples of the training data with replacement).

Histogram-based gradient boosting (XGB): it is a popular ensemble method relying on a boosting strategy. It minimizes the prediction loss by combining many decision tree regressors.

Additional models can be straightforwardly added and tested in the deployed version of the system. Notice that some models (e.g., Neural Networks) have not been included in this first release due to the lack of a sufficiently large amount of training data.

4.3 Methodology for old vehicles

Old vehicles are assumed to have a sufficiently large amount of historical data to train reliable Machine Learning models (see Section 2). Thus, separately for each vehicle we train the multiple regression models described in the previous section. Among the trained models, we select those that minimizes the mean residual error over the last 29 days predicting the maintenance ($E^v_{MRE}(\widetilde{D})$ with $\widetilde{D} = \{1, \ldots, 29\}$). For each vehicle, we consider the first 70% of their samples (N^v) as training set, and the remaining part as test set.

4.4 Methodology for new and semi-new vehicles

To handle new and semi-new vehicles, we need to face the following issues: (i) The lack of historical usage data, which hinders the training of per-vehicle regressors. (ii) The first maintenance cycle of most vehicles appears to have peculiar characteristics, with less usage. Indeed, the mean daily utilization time spent by the vehicles within the first cycle (10 676 seconds) is approximately 30% lower than in the subsequent cycles (13 792 seconds).

We design ad-hoc strategies to predict D^v for semi-new and new vehicles. We consider as training data the utilization series in the first cycle of many old vehicles. Collecting in the training set only usage data related to the first maintenance cycle allows Machine Learning models to focus on the usage patterns peculiar to that usage period (which could be significantly different all from the subsequent ones). We take 70% of the 24 vehicles (i.e., 17 vehicles), and consider their complete first cycle as training set. The first cycle of the remaining 30% of the vehicles (i.e., 7) is considered as test set.

Table 1: $E_{MRE}(\{1,...,29\})$ with models trained on all data and models trained in the last 29 days before maintenance, i.e., at times $i : D(i) \in \widetilde{D} = \{1,...,29\}$

Algorithm	Trained on all data $E_{MRE}(\{1,...,29\})$	Trained on $D = \{1,...,29\}$ $E_{MRE}(\{1,...,29\})$
BL	20.2	20.2
LR	26.1	10.8
LSVR	13.3	6.1
RF	6.9	2.4
XGB	10.9	5.6

4.4.1 Semi-new vehicles. When more than half of the first cycle has been completed, we learn a regression model by combining data from the other training vehicles. We apply the following strategies:

Baseline: we follow the same approach as for *old vehicles.* In practice, we compute AVG^v as the average utilization in the first half of the first cycle, i.e, $\sum_{t:\sum_{x\leq t} U^v(x)\leq T^v/2}(U^v(t)/|t : \sum_{x\leq t} U^v(x) \leq T^v/2)|)$.

Unified ML model: We create a single model, hereafter denoted as $Model_{Uni}$, on the first cycle data of the training vehicles. In this case, we train a single regression model for all the semi-new vehicles by merging data acquired from all the training vehicles together. The same model is applied to all the test vehicles.

Similarity-based ML model: We pick usage data only for the most correlated vehicle (rather than for all the old vehicles) and train vehicle-specific Machine Learning models, denoted as $Model_{Sim}$, on it. The key idea is to first decide whether each vehicle is similar to the target semi-new vehicles or not by estimating the pairwise correlation between the utilization series acquired in the first half of the first cycle. Then, we train the regression model only the first cycle data of the selected vehicle. In the current implementation, we estimate the pairwise similarity in terms of point-wise average distance AVG^v between the utilization series. However, more advanced similarity measures (e.g., [9]) can be integrated as well.

4.4.2 New vehicles. For these vehicles we have very few or none data at all, hence we cannot even compute AVG^v. Therefore, the baseline and similarity-based ML models cannot be applied. Hence, we apply the Unified ML model ($Model_{Uni}$). Notice that, when dealing with new vehicles, it does not make sense to compute $E^v_{MRE}(\{1,...,29\})$ since when we are approaching the deadline the vehicle will already be semi-new. Hence, we focus on comparing the algorithm performance in terms of the global error E^v_{Global}.

5 EXPERIMENTAL RESULTS

We have tested the proposed methodologies on real vehicles of different categories (new, semi-new, and old). The experiments were performed on a machine with Intel(R) Core(TM) i7-8750H CPU with 16 GB of RAM. To tune the algorithm parameter settings we have performed, separately for each vehicle, a grid search using a 5-fold cross validation. Specifically, for RF and XGB we have tuned the maximum tree depth from 3 to 50, and the number of estimators from 10 to 1000. For SVR, we tested the linear kernel and varied the values of the parameters epsilon (from 0.5 to 2.5) and C (from 0.01 to 100).

5.1 Results for old vehicles

In Table 1 we show the values of $E_{MRE}(\{1,...,29\})$ achieved on the test set by training each algorithm on the whole training data

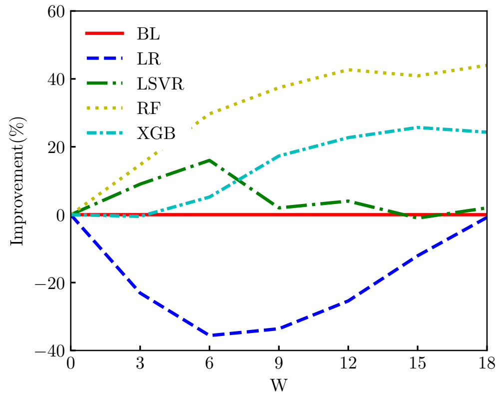

Figure 4: Improvement (%) for each algorithm by increasing the number of features. W is the window of past usage in the time series $U^v(t)$.

Table 2: Best setting for features and the corresponding mean relative error of the different algorithms.

Algorithm	Best window W	$E_{MRE}(\{1,...,29\})$
BL	0	20.2
LR	0	10.8
LSVR	6	5.2
RF	18	1.3
XGB	12	4.2

(central column) and just in the the last 29 days of the cycles in the training data (right hand-side column). $E_{MRE}(\{1,...,29\})$ is the average of the mean residual errors $E^v_{MRE}(\{1,...,29\})$ computed over all the test vehicles.

We found out that by forcing the algorithm to train only on the last 29 days, the error was reduced by 59% in LR, 54% in LSVR, 65% in RF and 48% in XGB. Since BL is not trained, its results do not change. In general, RF presents the best results, with an average relative error of only 6.9 and 2.4 days, respectively for the two training sets. Second best results are obtained by XGB, closely followed by LSVR.

Now we delve into the study of the usage of different features. Figure 4 shows, for each algorithm, the performance variation (percentage) by increasing the window size W. Positive variation means decrease of the error shown in Table 1 (i.e., performance improvement), and vice-versa. W is the window of past usage in the time series $U^v(x)$. W equal to 0 means we are in the univariate case, while $W > 0$ means we are in the multivariate case. For example, W equal to 3 means usage of $U(t-1)$, $U(t-2)$ and $U(t-3)$ in the regression (other than L(t)). BL is obviously constant, since it is not using past values of U. In RF and XGB, adding features greatly improve the results. RF and XGB seem to reach a plateau in performance with more than 15 previous value of utilization U. Hence, having more than a couple of weeks of previous data is enough to reach good performance. Respectively, the results improved by 44% and 25% for RF and XGB. In LR the best results are instead obtained without adding features. Finally, for LSVR the results improve by adding up to 6 features, but then decrease again. We report in Table 2 for each algorithm the final best results for $E_{MRE}(\{1,...,29\})$ and the optimal size of the window W. The non-linear algorithms (XGB and RF) reach again the best results.

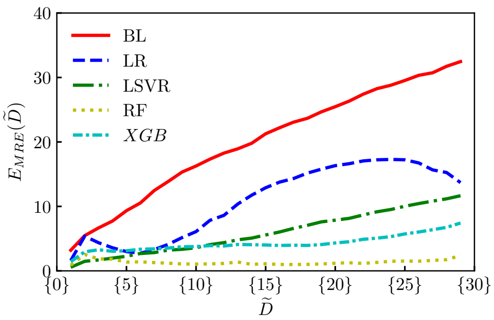

Figure 5: Mean relative error analyzed over all the test vehicles computed for $\widetilde{D}$ ranging from 1 to 29 days.

Finally we show in Figure 5 the error, i.e., $E_{MRE}(\widetilde{D})$ obtained with the best configuration, for each algorithm, in each of the last 29 days before maintenance. Here $\widetilde{D}$ is a set composed of a single value, corresponding to the number of days left for the the next maintenance. Clearly, the closer to the deadline, the smaller the error. All the algorithms improve the poor performance of the baseline. Interestingly, RF gets very good results even when we are 29 days from the deadline, with an average error of only 2.4.

We also analyze the time complexity of the proposed methodologies, in terms of execution time on our machine. The whole methodology pipeline includes (i) Data preparation, (ii) Model training, and (iii) Model Testing. Step (ii) was the most time-consuming task for all the algorithms. The average training time on a single vehicle is 30.4 s for XGB and 8.1 s for RF, while BL, LR, and LSVR are faster taking respectively 2.5 s, 3.8 s, and 2.8 s. Moreover, the model complexity increases more than linearly with the number of considered features, i.e., window size W.

5.2 Results for new and semi-new vehicles

Table 3 shows the results of the above discussed methods for semi-new vehicles. RF_{Sim} has the best results among all models, meaning that comparing the similarity of average usage can slightly improve the results (from 3.2 to 2.9 days in E_{MRE}). Moreover, notice how the baseline approach performs badly, with a mean relative error of 34.9, a value much higher than all the other metrics. This is because the limited amount of past data in the semi-new test vehicle cannot be trusted to create a simple predictor based on average usage.

In the last column of Table 3 we also report the results of the above discussed methods for new vehicles. Baseline and similarity-based models cannot be applied to new vehicle, since there is no past data. XGB_{Uni} has the best result among all models. Even if the results in terms of error appears poor, we recall that the global error accounts also for dates that are far from the deadline. Even more promising, the global error of XGB_{Uni} is comparable with the baseline mean relative error in the case of old vehicles (Table 2).

6 CONCLUSIONS AND FUTURE WORK

The paper has presented a Machine Learning application to support maintenance planning for fleets of industrial and construction vehicles. It proposes to use regression techniques to predict the remaining days until the next maintenance is due. The trained models are (i) vehicle-specific, when sufficient data is available to train reliable predictors, (ii) based on data acquired from similar vehicles, when the vehicle is semi-new (i.e., the first cycle maintenance is partly completed), or (iii) vehicle-independent, when the vehicle under analysis is new. The achieved results show that, when a minimal amount of vehicle-related data is available, Machine Learning approaches relying on non-linear models outperform both naive approaches and linear ML models.

The deployed version of the current system will be further extended. Specifically, we plan to enrich regression models using contextual information (e.g., meteorological data, fleet movements) and to design ML supported scheduling strategies.

Table 3: Results for semi-new and new vehicles

Algorithm	Semi-new vehicles $E_{MRE}(\{1,...,29\})$	New vehicles E_{Global}
BL	34.9	-
LR_{Sim}	4.9	-
$LSVR_{Sim}$	6.2	-
RF_{Sim}	2.9	-
XGB_{Sim}	5.3	-
LR_{Uni}	5.1	27.2
$LSVR_{Uni}$	8.8	27.8
RF_{Uni}	3.2	30.1
XGB_{Uni}	4.2	17.9

REFERENCES

[1] Carlos Alonso de Armiño, Miguel Ángel Manzanedo, and Álvaro Herrero. 2019. Studying Road Transportation Demand in the Spanish Industrial Sector Through k-Means Clustering. In *International Joint Conference SOCO'18-CISIS'18-ICEUTE'18*. Springer International Publishing, Cham, 387–396.

[2] Luis Barreto, Antonio Amaral, and Teresa Pereira. 2017. Industry 4.0 implications in logistics: an overview. *Procedia Manufacturing* 13 (2017), 1245–1252.

[3] T.G. Crainic and G. Laporte. 1998. *Fleet Management and Logistics*. Springer US.

[4] Zahid Halim, Rizwana Kalsoom, and Abdul Rauf Baig. 2016. Profiling drivers based on driver dependent vehicle driving features. *Applied Intelligence* 44, 3 (01 Apr 2016), 645–664.

[5] Karl Henrik Johansson, Martin Törngren, and Lars Nielsen. 2005. *Vehicle Applications of Controller Area Network*. Birkhäuser, Boston, MA, 741–765.

[6] Q. Li, S. Meng, S. Wang, J. Zhang, and J. Hou. 2019. CAD: Command-Level Anomaly Detection for Vehicle-Road Collaborative Charging Network. *IEEE Access* 7 (2019), 34910–34924.

[7] Dena Markudova, Elena Baralis, Luca Cagliero, Marco Mellia, Luca Vassio, Elvio Amparore, Riccardo Loti, and Lucia Salvatori. 2019. Heterogeneous Industrial Vehicle Usage Predictions: A Real Case. In *Workshop Proceedings of the EDBT/ICDT 2019 Joint Conference (March 26, 2019, Lisbon, Portugal)*.

[8] Emad Mohamed, Parinaz Jafari, Ming-Fung Francis Siu, and Simaan AbouRizk. 2017. Data-Driven Simulation-Based Model for Planning Roadway Operation and Maintenance Projects. In *Proceedings of the 2017 Winter Simulation Conference (WSC '17)*. IEEE Press, Article Article 272, 12 pages.

[9] Rodica Neamtu, Ramoza Ahsan, Elke A Rundensteiner, Gabor Sarkozy, Eamonn Keogh, Hoang Anh Dau, Cuong Nguyen, and Charles Lovering. 2018. Generalized dynamic time warping: Unleashing the warping power hidden in point-wise distances. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. IEEE, 521–532.

[10] Federico Perrotta, Tony Parry, and Luis C. Neves. 2017. Application of machine learning for fuel consumption modelling of trucks. In *2017 IEEE International Conference on Big Data, Boston*. 3810–3815.

[11] Mehdi Rashidnejad, Sadoullah Ebrahimnejad, and Jalal Safari. 2018. A bi-objective model of preventive maintenance planning in distributed systems considering vehicle routing problem. *Computers & Industrial Engineering* 120 (2018), 360 – 381.

[12] Elodie Robert, Christophe Berenguer, Keomany Bouvard, Hoceane Tedie, and Romain Lesobre. 2018. Joint dynamic scheduling of missions and maintenance for a commercial heavy vehicle: value of on-line information. *IFAC-PapersOnLine* 51, 24 (2018), 837 – 842.

[13] Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, and Vipin Kumar. 2018. *Introduction to Data Mining* (2nd ed.). Pearson.

[14] Dieter Uckelmann. 2008. A Definition Approach to Smart Logistics. In *Next Generation Teletraffic and Wired/Wireless Advanced Networking*, Sergey Balandin, Dmitri Moltchanov, and Yevgeni Koucheryavy (Eds.). Springer Berlin Heidelberg, 273–284.

[15] W. Zhang, D. Yang, and H. Wang. 2019. Data-Driven Methods for Predictive Maintenance of Industrial Equipment: A Survey. *IEEE Systems Journal* 13, 3 (Sep. 2019), 2213–2227.