Application of machine learning for improving contemporary ETA forecasting

Gabrielė Kasputytė[1,3,†], Arnas Matusevičius[1,3,†] and Tomas Krilavičius[2,3,*]

[1] Vytautas Magnus University, Faculty of Informatics, Department of Mathematics and Statistics, Vileikos street 8, LT-44404 Kaunas, Lithuania
[2] Vytautas Magnus University, Faculty of Informatics, Department of Applied Informatics, Vileikos street 8, LT-44404 Kaunas, Lithuania
[3] Centre for Applied Research and Development, Lithuania

Abstract

The popularity of road transport is growing ever higher in the modern day of age and is the most popular mode for transporting goods [1]. Usually, transportation increases the prime cost of the goods or services, thus, to increase a company's profit, effective vehicle routing becomes essential. In this research, we have analysed the possibility of improving the forecast of estimated time of arrival by ranking the drivers based on their behaviour data and estimating deviations from planned arrival time using different machine learning methods. The ranking was performed with TOPSIS and VIKOR methods, while the forecasting was performed using five machine learning algorithms: decision tree, random forest, XGBoost, Support Vector Machine and k-Nearest Neighbours. The performance of the forecasting models was evaluated using the adjusted coefficient of determination, root square mean error and mean absolute error metrics. It was concluded that the VIKOR method should be used to rank the drivers. Moreover, research results revealed that the best forecasting performance was achieved using an ensemble model based on random forest and Support Vector Machine models.

Keywords

Estimated time of arrival (ETA), freight transport, driver's score, ranking, machine learning, TOPSIS, VIKOR, Decision Tree, Random Forest, XGBoost, Support Vector Machine, k-Nearest Neighbours

1. Introduction

To improve the transparency of a supply chain, its participants use transport management systems as well as tracking and monitoring systems. Vehicles are getting equipped with stronger microprocessors, larger memory capacity and real-time operating systems. The newly installed technological platforms can use more advanced applications of the operating system, including model-based process control functions, artificial intelligence, and comprehensive computation.

Therefore, an increasing trend of implementing the latest developments in information and internet technologies in the transport sector as well as a tendency of developing new research-based systems that can quickly adapt to an ever-changing environment is observed. The goal of this study is to identify methods, best suited for solving the estimated time of arrival (ETA) forecasting problem.

The rest of the paper is organized as follows. Related work in this area is presented in Section 2. Section 3 introduces the datasets used in the current study. Section 4 presents the selected ranking and forecasting techniques. Experimental results are provided in Section 5. Finally, concluding remarks and future plans are discussed in Section 6.

2. Related work

To better understand what attributes and methods can be used for ranking drivers, research [2] was studied. The aim of this research was to categorize drivers according to their risk-proneness by analysing a GPS-based device's urban traffic data. Hierarchical Clustering Algorithm (HCA) and Principal Component Analysis (PCA) were used for the statistical analysis of the following driving parameters: *Speed over 60 km/h*, *Speed*, *Acceleration*, *Positive acceleration*, *Braking* and *Mechanical work*. The authors of this research conclude that while it is possible to classify the drivers according to these parameters, a lot of external factors, such as the driving environment or the condition of the assessed driver, are not taken into account.

A comparative analysis of two multi-criteria decision-making (MCDM) methods TOPSIS and VIKOR is presented in [3]. It was found that these two methods use different kinds of normalization (TOPSIS uses vector normalization, VIKOR – linear) to eliminate the units of criterion functions and use different aggregating functions for ranking. Nonetheless, both methods were found suitable for solving ranking problems.

IVUS 2022: 27th International Conference on Information Technology, May 12, 2022, Kaunas, Lithuania

[*] Corresponding author.

[†] These authors contributed equally.

gabriele.kasputyte@vdu.lt (G. Kasputytė);
arnas.matusevicius@vdu.lt (A. Matusevičius);
tomas.krilavicius@vdu.lt (T. Krilavičius)
0000-0001-8509-420X (T. Krilavičius)

The use of machine learning (ML) techniques for the ETA problem is discussed in [4] The authors applied artificial neural networks (ANNs) and support vector regression (SVR) to predict the time of arrival of container ships. Distance to the destination, the timestamp, geocoordinates, and weather information have been chosen as features. It was shown that SVR had performed better than ANNs and that the weather data did not have a significant impact on estimating the time of vessel arrivals.

In the [5] study, the k-Nearest neighbours (KNN), SVR, and the random forest algorithms were evaluated as methods for predicting the arrival time of open-pit trucks. A site-based approach was used as the position was only measured at a few discrete nodes of the route network. It was concluded that the random forest algorithm provides the best prediction results.

Ma et al. [6] proposed a tree-based integration method to predict traffic accidents by using different data variables. Predictions of the gradient boosting decision tree algorithm outperformed back propagation neural networks, support vector machines, and random forest. However, in this study, the nonlinear relationship between the influence characteristics and the predicted value was not analysed.

To improve travel time predictions, the author of [7] study applied the combination of random forest and gradient boosting regression tree (GBRT) models. The aim was to study how reducing a large volume of raw GPS data into a set of feature data affects high-quality travel time predictions. Only travel time observations from the previous departure time intervals were found to be beneficial features and were recommended by the author to be used as inputs when no other types of real-time information (e.g. traffic flow, speed) is available. Also, it is noted, that trees in GBRT models were found to be consistently much shorter than those of random forest models, leading to shorter computation times.

To sum up, characterizing attributes of drivers in research are usually derivative – data obtained from vehicle monitoring devices represent their driving behaviour. Since MCDM methods were found popular for conducting ranking procedures, two easily comparable MCDM methods will be used: TOPSIS and VIKOR. What concerns the ETA problem, the accuracy of the ML models tested in reviewed research was inconsistent. Therefore, a wide variety of ML methods suitable for the problem and available data will be evaluated. Namely, decision tree, random forest, XGBoost, Support Vector Machine (SVM) and KNN methods, as well as an ensemble of models, will be tested.

3. Research data

3.1. Data for ranking drivers

The available readings from a vehicle monitoring system that can be used to evaluate a driver's behaviour were extracted. The readings in this research cover the period from August 21, 2020, to January 1, 2022, and up to 398 observations representing different vehicles.

A dataset was constructed containing values for 7 attributes, namely *Free-rolling distance*, *Engine overloaded distance*, *Highest gear distance*, *Excess idling*, *Overspeeding time*, *Extreme braking events* and *Harsh braking events.*

3.2. Data for forecasting model

In this research, logistic transportation data was reviewed and a dataset was created for the ETA forecasting models. The initial dataset includes 1758 observations and 13 variables. The obtained information is from August 21, 2020 to January 24, 2022. A set of explanatory variables $\mathbf{X}$, with vectors $x_1, x_2, \ldots, x_{13}$, is obtained. The description of these variables is presented in Table 1.

Table 1
Set of explanatory variables

Variable	Description	Variable type
x_1	Driver's score	Ordinal
x_2	Tour beginning country	Categorical
x_3	Tour ending country	Categorical
x_4	Number of intermediate stops	Discrete
x_5	Furthest country	Categorical
x_6	Tour beginning month	Categorical
x_7	Tour beginning day	Categorical
x_8	Vehicle height	Continuous
x_9	Vehicle width	Continuous
x_{10}	Vehicle length	Continuous
x_{11}	Vehicle weight	Continuous
x_{12}	Hours of service breaks	Continuous
x_{13}	Planned distance	Continuous

Let T_i be the factual time when the ith cargo will be delivered, and t_i the planned time of delivery for the ith cargo. Then, the deviation from the planned time of delivery for the ith cargo Δt_i will be denoted as the difference between the planned and factual time of delivery:

$$\Delta t_i = T_i - t_i, \tag{1}$$

where $i = 1, 2, \ldots, n$. In that case, the explanatory variable is denoted as:

$$y_i = \Delta t_i. \tag{2}$$

This variable is the goal of the forecasting problem.

4. Methodology

4.1. VIKOR method

The VIKOR method was introduced as one applicable technique to implement within MCDM. It focuses on ranking and selecting from a set of alternatives in the presence of conflicting criteria, and on proposing a compromise solution (one or more). The compromise ranking algorithm VIKOR has the following steps [8]:

1. Determine the best f_i^* and the worst f_i^- values of all criterion functions, $i = 1, 2, \dots, n$. If the ith function represents a benefit then:
$$f_i^* = \max_j f_{ij}, \qquad f_i^- = \min_j f_{ij}. \tag{3}$$
2. Compute the values S_j and R_j, $j = 1, 2, \dots, J$, by the relations
$$S_j = \sum_{i=1}^{n} w_i \frac{f_i^* - f_{ij}}{f_i^* - f_i^-}, \tag{4}$$
$$R_j = \max_i \left[w_i \frac{f_i^* - f_{ij}}{f_i^* - f_i^-} \right], \tag{5}$$
where w_i are the weights of criteria, expressing their relative importance.
3. Compute the values Q_j, $j = 1, 2, \dots, J$, by the relation
$$Q_j = v \frac{S_j - S^*}{S^- - S^*} + (1 - v) \frac{R_j - R^*}{R^- - R^*}, \tag{6}$$
where
$$\begin{aligned} S^* &= \min_j S_j, \quad S^- = \max_j S_j, \\ R^* &= \min_j R_j, \quad R^- = \max_j R_j, \end{aligned} \tag{7}$$
and v is introduced as weight of the strategy of "the majority of criteria" (or "the maximum group utility"), here $v = 0.5$.
4. Rank the alternatives, sorting by the values S, R and Q, in decreasing order. The results are three ranking lists.
5. Propose as a compromise solution the alternative (a') which is ranked the best by the measure Q (minimum) if the following two conditions are satisfied:

   C1. "Acceptable advantage":
$$Q(a'') - Q(a') \geqslant DQ \tag{8}$$
   where a'' is the alternative with second position in the ranking list by Q; $DQ = 1/(J - 1)$; J is the number of alternatives.

   C2. "Acceptable stability in decision-making": Alternative a' must also be the best ranked by S or/and R. Here, v is the weight of the decision-making strategy "the majority of criteria".

4.2. TOPSIS method

The basic principle of the TOPSIS method is that the chosen alternative should have the shortest distance from the ideal solution and the farthest distance from the negative-ideal solution [9]. The TOPSIS procedure consists of the following steps:

1. Calculate the normalized decision matrix. The normalized value r_{ij} is calculated as
$$r_{ij} = \frac{f_{ij}}{\sqrt{\sum_{j=1}^{J} f_{ij}^2}}, \tag{9}$$
where $j = 1, \dots, J; \quad i = 1, \dots, n.$
2. Calculate the weighted normalized decision matrix. The weighted normalized value v_{ij} is calculated as
$$v_{ij} = w_i \cdot r_{ij}, \tag{10}$$
where $j = 1, \dots, J; \quad i = 1, \dots, n,$ w_i is the weight of the ith attribute or criterion, and $\sum_{i=1}^{n} w_i = 1$.
3. Determine the ideal and negative-ideal solution.
$$A^* = \{(\max_j v_{ij} | i \in I'), (\min_j v_{ij} | i \in I'')\}, \tag{11}$$
$$A^- = \{(\min_j v_{ij} | i \in I'), (\max_j v_{ij} | i \in I'')\}, \tag{12}$$
where I' is associated with benefit criteria, and I'' is associated with cost criteria.
4. Calculate the separation measures, using the n-dimensional Euclidean distance. The separation of each alternative from the ideal solution is given as
$$D_j^* = \sqrt{\sum_{i=1}^{n} (v_{ij} - v_i^*)^2}, \tag{13}$$
where $j = 1, \dots, J$.
Similarly, the separation from the negative-ideal solution is given as
$$D_j^- = \sqrt{\sum_{i=1}^{n} (v_{ij} - v_i^-)^2}, \tag{14}$$
where $j = 1, \dots, J$.
5. Calculate the relative closeness to the ideal solution. The relative closeness of the alternative a_j with respect to A^* is defined as
$$C_j^* = \frac{D_j^-}{D_j^* + D_j^-}, \tag{15}$$
where $j = 1, \dots, J$.
6. Rank the preference order.
Items C_j are ordered in descending order. The highest number indicates the best solution.

4.3. Decision Tree

Decision tree is a type of supervised learning algorithm that can be used in both regression and classification problems [10]. Each node is related to an attribute, whereas the leaves of the tree represent the final solution as the result of combining values of the attributes.

The splitting process is stopped after a particular stopping criterion is met. For example, a given threshold for the minimum number of observations left in a node being reached or a given threshold for the minimum change in the impurity measure not succeeding any more by any variable can be a stopping criterion [11].

Let L be the initial dataset made out of training samples with known dependent variable values. At first, the tree will be made of only a root node t_1 which represents the full set of variables. The objective is to split the nodes into two decision nodes until a terminal node is reached, for example splitting L into t_L and t_R, then splitting t_L and t_R into further sub-nodes until a stopping criterion is met [12].

4.4. Random Forest

Random forest is a ML algorithm that constructs a multitude of decision trees at training time. The main principle of constructing a random forest is that it is formed by combining solutions from binary decision trees made using diverse subsets of the original dataset and subsets containing randomly selected features from the feature set.

Constructing small decision trees that only have a few features takes up only a little of the processing time, hence the majority of such trees' solutions can be combined into a single strong classifier.

Steps for constructing a random forest as presented in [10] are as follows:

1. First, assume that the number of cases in the training set is K. Then, take a random sample of these K cases, and use this sample as the training set for constructing the tree.
2. If there are p input variables, specify a number $m < p$ such that at each node, m random variables out of the p can be selected. The best split on these m is used to split the node.
3. Each tree is subsequently grown to the largest extent possible and no pruning is needed.
4. Aggregate the predictions of the target trees to predict new data.
5. Finally, a decision is made by the majority rule.

4.5. XGBoost

XGBoost is a ML algorithm that implements frameworks based on Gradient Boosted Decision Trees [13]. XGBoost surpasses other ML algorithms by solving many data science problems faster and more accurately than its counterparts. Also, this algorithm has additional protection from overfitting.

The objective function to be optimized is given by

$$obj(\theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k), \qquad (16)$$

where n is the number of iterations, $l(y_i, \hat{y}_i)$ is the training loss function, $\hat{y}_i = \sum_{k=1}^{K}$ is the number of trees, Ω is the regularization term, $f_k \in \mathcal{F}$, and $\mathcal{F}$ is the set of possible classification and regression trees.

Writing the prediction value at step t as $\hat{y}_t^{(t)}$, gives

$$\hat{y}_t{}^{(t)} = \sum_{k=1}^{t} f_k(x_i) = \hat{y}_t{}^{(t-1)} + f_t(x_i). \qquad (17)$$

Next, a tree which optimizes our objective is chosen.

$$\begin{aligned} obj^{(t)} &= \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{i=1}^{t} \Omega(f_i) = \\ &= \sum_{i=1}^{n} l(y_i, \hat{y}_i{}^{(t-1)} + f_t(x_i)) + \Omega(f_t) + b, \end{aligned} \qquad (18)$$

where b is a constant.

To minimize the probability of overfitting, the complexity of the tree $\Omega(f)$ is defined as

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j{}^2, \qquad (19)$$

where

$$f_t(x) = w_{q(x)}, w \in R^T, q : R^d \rightarrow \{1, 2, \ldots, T\}. \qquad (20)$$

Here w is the vector of scores on leaves, q is a function assigning each data point to the corresponding leaf, and T is the number of leaves.

4.6. Support Vector Machine

The goal of SVM is to find the maximum separating hyperplane that would have the maximum distance between the nearest training data objects [14]. A separating hyperplane can be written as:

$$\mathbf{WX} + b = 0, \qquad (21)$$

where $\mathbf{W}$ is a weight vector, namely, $\mathbf{W} = \{w_1, w_2, \ldots, w_n\}$; $\mathbf{X}$ is a set of training data made of p number of objects, n number of attributes and an associated class label y_i; and b is a scalar constant.

The distance between hyperplanes, denoted as $2/||\mathbf{W}||$, has to be maximal. Consequently, this means that $||\mathbf{W}||$ (the Euclidean norm of the vector $\mathbf{W}$) has to be minimized. To simplify calculations, the Euclidean norm $||\mathbf{W}||$ can be swapped for $||\mathbf{W}||^2/2$. Thus, the objective function for this optimization problem is defined as:

$$\min_{\mathbf{W},b} \frac{1}{2}||\mathbf{W}||^2, \tag{22}$$

$$y_i(\mathbf{W}\mathbf{X}_i^T + b) \geq 1, \qquad i = 1, \ldots, p, \tag{23}$$

where the constraint (23) ensures that all objects from the training dataset will be positioned on the correct side of the appropriate marginal hyperplane.

The Lagrange multiplier strategy allows combining these two conditions into one:

$$\min_{\mathbf{W},b} \max_{\alpha \geq 0} \left\{ \frac{1}{2}||\mathbf{W}||^2 - \sum_{i=1}^{p} \alpha_i [y_i(\mathbf{W}\mathbf{X}_i^T - b) - 1] \right\}. \tag{24}$$

Kernel functions are used when the training dataset needs to be transformed into a higher-dimensional space due to the data being linearly inseparable.

$$K(\mathbf{X}_i, \mathbf{X}_j) = \phi(\mathbf{X}_i)\phi(\mathbf{X}_j)^T, \;\; \forall \mathbf{X}_i, \mathbf{X}_j \in \mathbf{X}. \tag{25}$$

In this study, the Gaussian radial basis function kernel was used:

$$K(\mathbf{X}_i, \mathbf{X}_j) = \mathrm{e}^{-\gamma||\mathbf{X}_i - \mathbf{X}_j||^2},$$

where the γ value is derived from the following equation:

$$\frac{1}{\gamma} \approx \mathrm{MED}_{i,j=1,\ldots,p}(||\mathbf{X}_i - \mathbf{X}_j||). \tag{26}$$

Here MED is the median. Usually parameter γ is found through trial and error.

4.7. k-Nearest Neighbours

The KNN algorithm is a method based on objects likeness [15]. In other words, the principle is to find the predefined number (k) of training samples closest to the new point. In the case of regression, the relationship between the explanatory variables and the continuous dependent variable is approximated by estimating the average of the observations, which together form the so-called neighbourhood. Its size is determined using cross-validation while minimizing the root mean square error.

The Euclidean distance was used to calculate the distance between objects.

4.8. Model evaluation metrics

The significance of regression in a model is usually calculated using the coefficient of determination [16]:

$$R_{yx_1x_2\ldots x_k} = \sqrt{1 - \frac{\sigma^2{}_{res}}{\sigma^2{}_y}}, \tag{27}$$

where $\sigma^2{}_{res}$ is a residual dispersion from a forecast model, $\sigma^2{}_y$ – dispersion of y.

However, the adjusted coefficient of determination $R^2{}_{adj}$ is better suited for comparing regression models as it avoids the inaccuracy, caused by numerous factors in the coefficient of determination [16]:

$$R^2{}_{adj} = 1 - (1 - R^2) \cdot \frac{n-1}{n-k-1}, \tag{28}$$

where n is the number of observations available for analysis, k is the number of variables.

Moreover, statisticians are used to measuring accuracy by computing mean square error (MSE), or its square root conventionally abbreviated by RMSE (for root mean square error). The latter is in the same units as the measured variable and so is a better descriptive statistic. Moreover, it is the most popular evaluation metric used in regression problems. RMSE follows an assumption that errors are unbiased and follow a normal distribution. RMSE metric is given by [17]:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(S_i - O_i)^2}, \tag{29}$$

where O_i are the observations, S_i are the predicted values of a variable.

Moreover, the average magnitude of the forecast errors can be measured by the mean absolute error:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|S_i - O_i|. \tag{30}$$

In this case, the direction of errors is not being considered [17].

There are various ways to improve models depending on the technique involved. The most popular way is to construct ensemble models. Once there are multiple models that produce a score for a particular outcome, they can be combined to produce ensemble scores. For example, a new score can be calculated as the average of two classifiers and then assess it as a further model. Usually, the area under the curve improves for these ensemble models.

5. Results

5.1. Driver ranking

Selection of attribute weights. To begin the ranking procedure, first attribute weights had to be established.

Since the drivers must be ranked in compliance with attribute priorities dictated by a company, their importance was evaluated by an expert on a scale from 1 to 10. This is presented in Table 2. Thus, we get the first set of weights:

Table 2
The first set of weights

Attribute	Score	Weight	min/max
Free rolling distance	5	0.09	max
Engine overloaded distance	10	0.19	min
Highest gear distance	7	0.13	max
Excess idling	10	0.19	min
Overspeeding time	10	0.19	min
Extreme braking events	8	0.15	min
Harsh braking events	4	0.07	min

$$W_1 = w_1 + 3w_2 + w_3 + w_4 + w_5 = 1,$$

where $w_1 = 0.09$ is the weight of *Free rolling distance*, $w_2 = 0.19$ is the weight of *Engine overloaded distance*, *Excess idling* and *Overspeeding time*, $w_3 = 0.13$ is the weight of *Highest gear distance*, $w_4 = 0.15$ is the weight of *Extreme braking events*, $w_5 = 0.07$ is the weight of *Harsh braking events*.

However, since the importance of attributes can be biased, a baseline weight model was also tested.

$$W_2 = 7w_1 = 1,$$

where $w_1 = 1/7$ is the weight of each attribute.

Results of ranking methods. Ranking of the drivers was performed using the generated sets of weights. Criterion values, computed by TOPSIS (C_i) and VIKOR (Q) methods, were used to rank the drivers. However, in many instances the difference between two values of the same criteria had been minute, hence the values were grouped. Values were grouped using a ten-point system.

Considering the results of different ranking methods presented in Table 3, the method with the most logical ranking of the drivers was confirmed to be with the VIKOR method. In addition, the first weight set should be used when creating a dataset for the forecasting models, since it would comply with the attribute priorities from the company and no significant difference was observed between the two tested sets of weights.

Table 3
Score distribution with different weight sets

	TOPSIS		VIKOR	
Score	W_1	W_2	W_1	W_2
10	325	316	156	148
9	62	71	107	120
8	6	7	67	63
7	4	3	28	31
6	0	0	16	12
5	0	0	6	8
4	1	1	9	6
3	0	0	3	5
2	0	0	5	4
1	0	0	1	1

5.2. Forecasting models

For improving the forecast of ETA, it was enough to forecast deviation from planned duration, because this variable had already been computed by routing service. In that case, the goal was to forecast the deviation from planned tour duration. Overall five ML methods were tested: decision tree, random forest, XGBoost, support vector machine (SVM) and k-Nearest neighbours (KNN).

Quantitative variables were normalized using min-max normalization, while the categorical variables were transformed and added to the models by replacing them with binary dummy variables.

Therefore, when applying the random selection and assignment of the set indices to the test and training sets, 75% of the dataset was assigned to the training sample and 25% to the test sample. Cross-validation was used for the selection of optimal parameters in all five models. During this procedure in the regression models, the sample data was divided into 10 groups.

Further, the optimal parameters of all models are determined:

1. **The decision tree model.** The minimum number of observations that can be in a node was set to be seven. Furthermore, if a node is to be split, the minimum number of observations per node has to be 20.
   A total of 11 splits were made. The *hours of service breaks*, the *tour beginning day*, the *beginning*, *ending* and *furthest countries*, as well as the *planned tour distance* impacted the creation of the model.
2. **The random forest model.** The optimal number of randomly selected variables in each division of the random forest was set to 59 (this value had been determined based on a precision measure). Whereas the number of trees is a basic size of 500.
3. **The XGBoost model.** An optimal model is determined by the lowest value obtained for the RMSE error. The maximum tree depth value of 0.3 was obtained. The higher this value is, the more complicated the model becomes. Also, the ratio of partial sample training cases is 0.75. In other words, the XGBoost method randomly selects 75% of the training dataset prior to growing the trees, which in return protects from an over-

load of data. Such partial selection of a sample occurs once per each iteration. The maximum number of repetitions was determined to be 150.

4. **The k-Nearest neighbour model.** The k parameter for KNN model was established to be equal to 4. This value was selected by changing the parameter value from 1 to 10 and determining which parameter has the smallest RMSE value. The Euclidean distance measure was used to estimate the distance between the points.
5. **The SVM model.** The number of support vectors was established to be 1008 and the Gaussian radial basis function kernel was used.

The accuracy of all five constructed models was evaluated by predicting values of deviation from planned tour time for unseen test set data. The adjusted coefficient of determination R^2_{adj}, RMSE, and MEA were calculated to determine and compare the suitability of the prediction models. The results are presented in Table 4. It can be seen that for all three accuracy measures, the best results for predicting test data were obtained using a random forest model, where the mean absolute difference between the predicted and actual values had been almost 684, the square root of the average squared differences between the predicted and actual values had been 1 120.15, and the adjusted coefficient of determination had been 77.57%. XGBoost model yielded quite similar results, where R^2_{adj} had been lower by only 3.3%, the MAE error had been higher by 93.24 units and the RMSE had been higher by 90.83. The worst prediction results were obtained using KNN method, for which R^2_{adj} had been less than 25%.

Table 4
The accuracy of regression models

Model	$R^2{}_{adj}$	RMSE	MAE
Decision tree	0.6666	1391.18	792.56
Random forest	0.7757	1120.15	683.94
XGBoost	0.7427	1210.98	777.18
SVM	0.6726	1408.97	867.82
KNN	0.2465	2105.53	1208.22

A possibility to improve the models by forming an ensemble model was observed, hence a decision was made to try a combination of predictions from two models: the random forest and SVM. Several combinations were made. The first method estimated the average of the forecasts of both models:

$$\hat{y}_i = \frac{\hat{y}_{i_{rf}} + \hat{y}_{i_{svm}}}{2}, \tag{31}$$

where $\hat{y}_i$ is the predicted value of the ith observation for the model ensemble, $\hat{y}_{i_{rf}}$ are the predicted values of the ith observation of the random forest model and $\hat{y}_{i_{svm}}$ – of the SVM model. The adjusted coefficient of determination of this ensemble model resulted in 0.7672.

Another way to form an ensemble of models is by using the weighted sum method. In this type of ensemble, the prediction value of the better model (in this case the random forest model) is determined to have a weighting coefficient c_1, that is less than 1, but not less than 0.5. However, the total amount of weights must be equal to one, therefore, the weighting coefficient of the other model (SVM model) c_2 shall be greater than zero, but less than 0.5. Then, the new predicted value could then be obtained as follows:

$$\hat{y}_i = c_1 \cdot \hat{y}_{i_{rf}} + c_2 \cdot \hat{y}_{i_{svm}}. \tag{32}$$

The equation

$$c_1 = 1 - c_2 \tag{33}$$

must be met, thus (32) can be written as:

$$\hat{y}_i = (1 - c_2) \cdot \hat{y}_{i_{rf}} + c_2 \cdot \hat{y}_{i_{svm}}. \tag{34}$$

In order to find with which weight the adjusted coefficient of determination of the ensemble model obtains the highest value, the value of c_2 was being changed from 0.01, 0.02, 0.03, and so on to 0.49. The experiment resulted in a maximum value of R^2_{adj} (0.7795) when c_2 was equal to 0.2. The second way of constructing an ensemble model resulted in a higher R^2_{adj} than the first method, hence, the second was more suitable. Therefore, the expression of the final ensemble model was as follows:

$$\hat{y}_i = 0.8 \cdot \hat{y}_{i_{rf}} + 0.2 \cdot \hat{y}_{i_{svm}}. \tag{35}$$

The forecast graph of the created ensemble model is presented in Fig. 1. Some outliers remained poorly predicted, but the overall prediction is accurate. Metrics evaluating the created ensemble model are presented in Table 5. In comparison to the results of individual models (Table 4), higher accuracy could be observed in all three metrics of the ensemble model. Nonetheless, the improvement in accuracy was not significant: the adjusted coefficient of

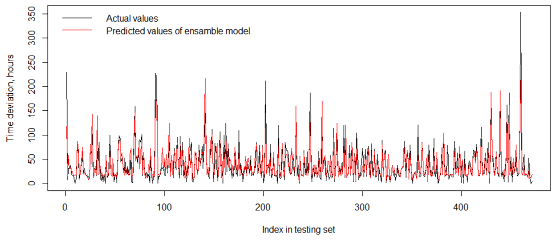

Figure 1: Predicted values of the model ensemble

determination was higher than the best individual model by only 0.38%, the RMSE was lower by 1.87, and MAE was lower by 0.47.

Table 5
The accuracy of the model

Model	$R^2{}_{adj}$	RMSE	MAE
Ensemble model	0.7795	1118.28	683.47

6. Conclusions

In this research, in order to improve the forecast of contemporary ETA, the possibility to rank the drivers based on their behaviour data and predict deviations from planned arrival time using different ML methods were analysed. For this purpose, a dataset consisting of vehicle monitoring data was used for ranking the drivers with TOPSIS and VIKOR methods. It was found that the results of the VIKOR method with the company's attribute importance weight set produced the most suitable drivers' scores. Then, these scores were used to supplement a new dataset constructed for ML methods. Moreover, five methods: decision tree, random forest, XGBoost, SVM, KNN, were used to create the deviation from the planned tour duration forecasting model. Finally, the ensemble model based on the random forest and SVM resulted in the most accurate results ($R^2_{adj} = 77.95\%$).

In the future, it is planned to continue the construction of the improved ETA prediction model by including real-world parameters that a vehicle takes into account while driving a certain route. For example, the need to stop for mandatory driving breaks or filling up would be considered.

Acknowledgments

We thank Tadas Motiejūnas, Darius Ežerskis, Viktorija Naudužaitė, Donatas Šumyla and UAB Ruptela (https://www.ruptela.com/) for cooperation and useful insights.

References

[1] Z. Wang, K. Fu, J. Ye, Learning to estimate the travel time, 2018, pp. 858–866. doi:10.1145/3219819.3219900.

[2] Z. Constantinescu, C. Marinoiu, M. Vladoiu, Driving style analysis using data mining techniques, International Journal of Computers, Communications & Control (IJCCC) V (2010) 654–663. doi:10.15837/ijccc.2010.5.2221.

[3] S. Opricovic, G.-H. Tzeng, Compromise solution by mcdm methods: A comparative analysis of vikor and topsis, European Journal of Operational Research 156 (2004) 445–455. URL: https://www.sciencedirect.com/science/article/pii/S0377221703000201. doi:https://doi.org/10.1016/S0377-2217(03)00020-1.

[4] I. Parolas, Eta prediction for containerships at the port of rotterdam using machine learning techniques, 2016.

[5] X. Sun, H. Zhang, F. Tian, L. Yang, The use of a machine learning method to predict the real-time link travel time of open-pit trucks, Mathematical Problems in Engineering 2018 (2018) 1–14. doi:10.1155/2018/4368045.

[6] X. Ma, C. Ding, L. Sen, Y. Wang, Y. Wang, Prioritizing influential factors for freeway incident clearance time prediction using the gradient boosting decision trees method, IEEE Transactions on Intelligent Transportation Systems 18 (2017) 2303–2310. doi:10.1109/TITS.2016.2635719.

[7] X. Li, R. Bai, P. O. Siebers, C. Wagner, Travel time prediction in transport and logistics: Towards more efficient vehicle gps data management using tree ensemble methods, VINE Journal of Information and Knowledge Management Systems 49 (2019) 277–306. doi:10.1108/VJIKMS-11-2018-0102.

[8] D. Servaitė, R. Juozaitienė, T. Krilavičius, Logistics service provider selection using topsis and vikor methods, 2020. URL: https://www.vdu.lt/cris/bitstream/20.500.12259/110824/2/ISSN1613-0073_2020_V_2698.PG_86-91.pdf.

[9] S. Opricovic, G.-H. Tzeng, Compromise solution by mcdm methods: A comparative analysis of vikor and topsis, European Journal of Operational Research 156 (2004) 445–455. URL: https://www.sciencedirect.com/science/article/pii/S0377221703000201. doi:https://doi.org/10.1016/S0377-2217(03)00020-1.

[10] J. Le, Decision trees in r, DataCamp (2018). https://www.datacamp.com/community/tutorials/decision-trees-R.

[11] C. Strobl, J. Malley, G. Tutz, An introduction to recursive partitioning: Rationale, application, and characteristics of classification and regression trees, bagging, and random forests, Psychological methods 14 (2009) 323–48. doi:10.1037/a0016973.

[12] G. Norkevičius, G. Raškinis, Garsų trukmės modeliavimas, klasifikavimo ir regresijos medžiais naudojant didelės apimties garsyną, in: Informacinės technologijos 2007, 2006, pp. 52–66.

[13] xgboost developers, xgboost Release 1.2.0-SNAPSHOT, 2020. URL: https://buildmedia.readthedocs.org/media/pdf/xgboost/latest/xgboost.pdf.

[14] J. Han, M. Kamber, J. Pei, Data Mining: Concepts and Techniques, Second Edition (The Morgan Kaufmann Series in Data Management Systems), 2 ed., Morgan Kaufmann, 2006.

[15] Murni, R. Kosasih, A. F. Oeoen, T. Handhika, I. Sari, D. Lestari, Travel time estimation for destination in bali using knn-regression method with tensorflow, IOP Conference Series: Materials Science and Engineering 854 (2020) 012061. doi:10.1088/1757-899X/854/1/012061.

[16] H. Altland, Regression analysis: Statistical modeling of a response variable, Technometrics 41 (2012) 367–368. doi:10.1080/00401706.1999.10485936.

[17] C. Chatfield, Time-series forecasting, Significance 2 (2005) 131–133. URL: https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1740-9713.2005.00117.x. doi:https://doi.org/10.1111/j.1740-9713.2005.00117.x.