Improving the quality of the driving cycle by processing statistical data of vehicle movement

Maksim Potashnikov¹*, Alexander Muravev¹, Alexander Kartashov¹, Nikita Pikalov¹, and Sergey Nazarenko²

¹Bauman Moscow State Technical University, Moscow, 2nd Baumanskaya st., 105005, Russia
²PJSC «KAMAZ”, Transportniy proezd, 420000, Naberezhnye Chelny, Republic of Tatarstan, Russia

Abstract. The article describes the process of analyzing and processing statistical data on the movement of freight transport obtained using navigation equipment installed on the vehicles in question. A detailed analysis of the source data, as well as all stages of their processing, was carried out using specialized Python tool packages such as Pandas, Numpy and Sklearn. The effectiveness of the proposed approach to data processing is shown by comparing the driving cycles of vehicles generated on the basis of the original and modified data. The topic of data analysis and processing is not new, but the relevance of the article lies in the successful application of common and effective processing methods to statistical data on the movement of vehicles, work with which is complicated by the fact that they are obtained using widespread, but focused on other tasks equipment. Thus, the work opens up new opportunities for statistical analysis of the movement of freight transport without the need to replace navigation equipment.

1 Introduction

In modern world, working with big data is an integral part of many areas of science and production; every year data analysis becomes necessary to solve an increasing number of new tasks. Mechanical engineering is no exception since a large number of modern design and verification calculations along with virtual simulation tests are carried out on the basis of statistical data analysis [1].

In [4-7], it is shown how statistical data can be used to generate driving cycles corresponding to traffic conditions in different regions, with the aim of their further application in determining load modes for transmission units and aggregates [2-3]. In the future, such an approach can significantly improve the adaptability of vehicles to specific operating conditions and thus increase their service life.

To monitor vehicle dynamics and collect statistical data, various telematics equipment is used. This equipment includes navigation terminals connected to the vehicle's on-board CAN-bus and equipped with GLONASS/GPS receivers. The main problem of using statistical information obtained with this type of equipment is its poor quality: the data sets

* Corresponding author: potashnikov@bmstu.ru
obtained have a large number of missing values, alongside with low measurement resolution or a relatively large interval between measurements.

It is worth noting that the data obtained with the help of this type of equipment meets all the requirements and provides complete solution to the problems of improving the efficiency of truck fleet operation and costs reduction of enterprises in various industries, which are cargo and passenger transportation, mining, construction and many other spheres [8-11]. The quality of the data obtained, though, is insufficient for the tasks of data analysis. This problem of data quality can be solved in two ways: by using expensive specialized equipment or by using different data processing algorithms. The use of specialized equipment for a large number of vehicles is difficult, both from a financial point of view and due to the need for periodic monitoring of measurements by a technical specialist. Therefore, the most preferable option is to use various processing algorithms that allow to transform vehicle traffic statistics into a form that is satisfactory for further use in generating driving cycles and conducting virtual vehicle tests.

2 Analysis of source data

This paper uses statistical data on the movement of trucks with a maximum weight equal to 11.5 tons per axle, in particular, KAMAZ-54901 mainline tractors and KAMAZ-5325 trucks. Data was collected from 29 machines. The time period of records varies from 1.5 months to 1 year. The data set contains the following information (columns):

- "time" – time in the format YY-MM-DD HH:MM:SS.
- "latitude" and "longitude" – coordinates of the vehicle at the current time.
- "altitude" – the height of the vehicle above sea level.
- "speed" – the speed of the vehicle calculated on the basis of data from the GPS receiver, expressed in km/h.
- "axle_weight" – the weight that falls on the driving axle of the vehicle.
- "high_resolution_total_vehicle_distance_resolution_total_vehicle_distance" – vehicle mileage, expressed in kilometers to the nearest meter.
- "engine_speed" – internal combustion engine rotation speed.
- "high_resolution_engine_total_fuel_used", "engine_total_fuel_used" – fuel consumption.
- “wheel_based_vehicle_speed” – vehicle speed, calculated based on signals from wheel speed sensors.

The data set contains more than 100 million of records, but not all of them are informative due to the large number of missing values of the measured parameters. Table 1 shows the percentage of missing values from the total number of measurements of the corresponding parameters.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Percentage of missing values, %</th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td>0</td>
</tr>
<tr>
<td>latitude</td>
<td>0</td>
</tr>
<tr>
<td>longitude</td>
<td>0</td>
</tr>
<tr>
<td>altitude</td>
<td>11</td>
</tr>
<tr>
<td>speed</td>
<td>11</td>
</tr>
<tr>
<td>axle_weight</td>
<td>86.4</td>
</tr>
<tr>
<td>engine_speed</td>
<td>32.9</td>
</tr>
<tr>
<td>high_resolution_engine_total_fuel_used</td>
<td>52.8</td>
</tr>
<tr>
<td>high_resolution_total_vehicle_distance</td>
<td>41.3</td>
</tr>
<tr>
<td>engine_total_fuel_used</td>
<td>99.3</td>
</tr>
</tbody>
</table>
In the common practice of data analysis and machine learning [12], it is generally assumed that if a column contains more than 5-10% of missing values, then such data cannot be restored and is uninformative [11-3]. In some cases, though, data sets with a large number of missing values can also be useful and their use in analysis can have a significant impact on the final result.

When processing, first of all, it is necessary to get rid of data that is not of interest in the analysis or is uninformative and cannot be restored. When forming the load mode for electromechanical transmission uninformative units are the values of the internal combustion engine speed and fuel consumption ("engine_speed_speed", "high_resolution_engine_total_fuel_used_resolution_engine_total_fuel_used", "engine_total_fuel_used_total_fuel_used"). Also, the obtained data on vehicle speed values measured based on the wheel speed sensors ("wheel_based_vehicle_speed_based_vehicle_speed") have a large number of gaps, which cannot be restored. The data set that will be used for further monitoring is shown in Figure 1.

Fig. 1. Data set fragment

3 Processing speed, altitude and axle load data

It is possible to correct missing values without any data correction in columns containing speed and axle load values. When deleting rows in a dataset, it is important to make sure that no useful information is lost.

Analyzing the original data set (Figure 1), it can be noted that the missing values in the "speed" column appear when the navigation controller is disabled. In other situations, there are no missing values in this column. It also turned out that the "altitude" and "speed" columns have the same number of missing values, from which we can conclude that they appear at the same time, that is, when the vehicle is stopped and the power-off process occurs. Parts of the data where the vehicle is stationary, are not of interest for analysis, so rows with missing values in the speed and altitude columns were removed from the data set.

Analysis of the column with information about the axle load showed that it contains 86.4% of the missing values (Table 1). However, it is undesirable to completely remove it from consideration, since it is useful for a large number of calculations, for example, in
calculating the braking torque of a vehicle. Looking at the data in the axle load column (Figure 1), it can be seen that the sensor that measures the axle load sends a new value only when the axle load changes. Accordingly, missing values indicate that there are no changes in the axle load. In other words, the missing values in this column contain information that the axle load value has not changed, so you can't delete them. To get rid of omissions, it is enough to replace each missing value in the column with the nearest upper omitted value. In this case, there will be no loss of information.

As a result of performing the actions described above, all missing values in the dataset were deleted, with the exception of the column with mileage.

Another important feature in calculations and virtual tests is the current acceleration of the vehicle. For an approximate calculation of instantaneous acceleration, it is enough to know the speed difference and the time difference between two adjacent records in the data set. Speaking differently, the calculation uses the following formula:

\[
\alpha = \frac{\Delta V}{\Delta t},
\]

where \(\alpha\) is the current vehicle acceleration, \(\Delta V\) is the speed difference between two data set entries, and \(\Delta t\) is the time difference between two data set entries.

4 Processing mileage and time data

Time data analysis should be based on three columns: time, current speed, and mileage. The data fragment in the figure 1 shows two problems at once: missing data in the mileage column and data duplication in the time column. Presumably, this situation is caused by poor synchronization of measuring devices, as well as possible loss or neglect of a tenth of a second when recording the current time.

To solve this problem, one can delete all duplicating rows in the time column, while keeping the rows where the mileage value is not missing, and then fill in the small number of remaining gaps in the mileage column by adding the distance obtained based on the current speed and time difference in the corresponding rows to the previous mileage value. However, this option is undesirable, as a large amount of useful information will be lost. In this case, approximately 60% of all rows in the data set will be deleted, and a significant amount of mileage information will also be lost, since rows with repeated time data often contain different mileage values (1017 and 1018 rows in Figure 1). Moreover, a lot of time information will be lost, because after the processing described above, lines will often appear where the time value differs by 2-3 seconds, which obviously does not correspond to reality.

It is more efficient to consider and process sections with repeated time data separately, without deleting rows from the data set, but correcting them. When analyzing the data in detail, it was concluded that the mileage measurement device sends values once per second, provided that the vehicle speed is greater than 18 km/h or 5 m/s. If the vehicle moves at a speed less than 5 meters per second, the mileage counter is not updated, and in this case it does not send a value. In this regard, the processing algorithm should be divided into two parts: for data sections where the speed is greater than 18 km/h and for sections where the speed is less than 18 km / h. Basing on this decision, the corresponding algorithm was developed, the block diagram of which is shown in Figure 2.
The reference value in this case will be the value of the mileage. By looking at the dynamics of the change in mileage, one will notice that it changes with an accuracy of 5 meters, and also that in the fragment in Figure 1, its values change by 15-20 meters each iteration at a constant speed of 62-63 km/h. Assuming that the mileage meter sends data at a fixed interval, you can calculate this interval by calculating the formula 2.

\[ \Delta t = \frac{\Delta S}{V_{sec}} = \frac{17.5}{62.5/3.6} \approx 1 \text{sec}, \]  

(2)
where $\Delta t$ is the time interval between entries in the mileage column, $\Delta S$ is the path traveled during this interval (the difference between adjacent non-missed values in the mileage column), and $v_{curr}$ is the corresponding current vehicle speed.

As a result of the algorithm, all missing mileage values and repeated time data were eliminated without significant data loss and distortion.

5 Processing height data

Referring to the "altitude" column in Figure 1, we can see that the height changes "stepwise", and for further processing of the graph, this step change must be smoothed out. When developing the algorithm for smoothing height values, it was assumed that in areas where the height values are repeated, it changes linearly. In other words, the algorithm works by searching for sections of repeated data and replacing them with linearly increasing or linearly decreasing ones, depending on the first non-repeating value after each section. The result of the smoothing algorithm is shown in Figures 3 (a) and 3 (b).

![Fig. 3](image)

**Fig. 3** – Graphs of the dependence of the vehicle height on time: a – before processing altitude data; b – after processing altitude data

6 Calculating the slope angle of the road

Based on the available data, it is possible to calculate the slope of the road at the current time. The slope value is an important parameter, for example, when calculating loads on transmission units and aggregates [3]. Due to the available qualitative data on vehicle mileage and altitude (processed in the previous paragraphs), the slope of the road can be calculated using formula 3, similar to how similar calculations were performed in [14].

$$\alpha = \arcsin \left( \frac{\Delta h}{\Delta S} \right),$$  \hspace{1cm} (3)

where $\alpha$ is the current road slope, $h$ is the height difference between two entries in the data set, and $\Delta S$ is the mileage difference between two entries in the data set.

There is an alternative way to calculate the road slope using the vehicle's longitude and latitude values, using the Haversine formula (Formula 4). If this formula is used, there is no need to process the vehicle mileage data, since it is not used in the formula.
\[ \sin(\alpha) = \frac{\Delta h}{2R\arcsin\left(\frac{\Delta \phi}{2} + \cos(\varphi_1) \cos(\varphi_2) \sin^2\frac{\Delta \lambda}{2}\right)} \]  

(4)

where \( \Delta \varphi = \varphi_2 - \varphi_1 \) is the difference in latitudes, \( \Delta \lambda = \lambda_2 - \lambda_1 \) is the difference in longitudes, and \( R \) is the Earth's radius.

Let us compare two calculated road gradients. Figure 4a shows the slope calculated using formula (2) using the real vehicle mileage column, and Figure 4b shows the slope calculated using formula (3).

![Graph of the dependence of the road slope on time: a – obtained on the basis of real mileage records; b – obtained on the basis of calculations using the Haversine formula.](image)

**Fig. 4.** Graph of the dependence of the road slope on time: a – obtained on the basis of real mileage records; b – obtained on the basis of calculations using the Haversine formula

The graphs show that the angle calculated on the basis of the processed mileage records changes more smoothly and has significantly fewer fluctuations, which is a significant advantage compared to the slope data calculated using the Haversine formula. Formula 4 gives this result due to the fact that the latitude and longitude data in the data set are also not accurate enough and their values are fluctuating, which affects the calculation of the slope. This can be dealt with by filtering the slope values using, for example, a Kalman filter or a moving average. However, the filtered data will still be inferior to the calculation using formula 3, and will also be shifted in time due to the features of the filters used.

7 Converting data for use in the ride cycle generation method

In [2], it is shown that high-quality driving cycles of vehicles can be generated based on traffic statistics, but it does not disclose how the data is pre-processed in order to let the driving cycle meet the convergence requirements for a number of parameters with a complete real-world record of vehicle movement. The current section shows that the correct preparation of data for such algorithms is often the determining factor for success.

To generate cycles, we will use the "micro-trip" method using clustering [2]. Accordingly, it is necessary to select "micro-trips" from the data set and then assign them certain criteria, similar to the work [2], by which they will be selected for participation in the final driving cycle. To select a separate "micro-trip", it is enough to find the start and end indices in the general data set.

After splitting the data set into "micro-trips", they need to be characterized by several parameters in order to select the "micro-trips" included in the driving cycle, not by chance,
but based on the analysis of these very parameters. If we add the corresponding parameters, we get a data set, each row of which corresponds to one "micro-trip" and contains the parameter values for each "micro-trip".

Looking at the correlation matrix (Figure 5) for the parameters, it can be seen that the average deceleration strongly correlates with the average acceleration, and the length of the "micro trip" strongly correlates with its duration. Since clustering will then be used to select "micro-trips", one should get rid of correlation, since almost all machine learning methods, including clustering methods, work very poorly with data that have strongly correlated features. Therefore, the average deceleration and micro-trip time columns should be excluded from consideration.

![Correlation Matrix](https:// DOI.org/10.1051/e3sconf/202346006029)

**Fig. 5.** Correlation matrix for parameters characterizing "micro-trips"

Also, when working with data, it is desirable that the features have a normal distribution. This makes it easier for machine learning methods, including clustering methods, to work, since basically such methods can only "notice" linear dependencies in the data. One of the most effective methods of data normalization is the Yeo-Johnson method [11-5]. The difference in data distributions before and after normalization can be seen in Figures 6a and b.
The next procedure, without which clustering methods will not work adequately is data scaling. It is very important that all features take on values only within the same numerical interval, otherwise when clustering algorithms work, a feature with a larger scale will have the greatest impact compared to other features.

Also, the data often contains so-called "outliers", i.e. data that stands out sharply from the rest. Such data are most often errors in measurements or calculations and should be eliminated so that they do not distort the results.

For scaling, we will use the median scaling algorithm, which performs two functions at once: scaling data and eliminating outliers (by "cutting off" the tails of the distribution). It can be provided by the following conversion formula:

$$x' = \frac{x - \text{median}(x)}{IQR}$$

$$IQR = Q3(x) - Q1(x),$$

where $x$ – initial data, $x'$ - scaled data, $IQR$ – difference between the 1st and 3rd quartiles of the trait distribution.

The results of scaling application are shown in Figure 7.
8 Generating a driving cycle

After preparing the data, it can be passed to the clustering algorithm and, basing on the results obtained, generate a ride cycle. In this paper, the "k-means" clustering method is used [2]. With its help, "micro-trips" are divided into clusters, from which they are selected to form the final driving cycle.

Since the goal of generating driving cycles is to statistically match the obtained cycle with a complete record of vehicle movement, we compared driving cycles generated from the processed and unprocessed data with the original statistical data in order to assess the quality of data before and after processing, using the following criteria [2]:
- average speed (km/h).
- average acceleration (m/s²).
- average deceleration (m/s²).
- travel time (seconds).
- ride length (m).
- the number of slopes over 5%.

The initial selection of a set of criteria was based on the recommendations according to [16] and included more points, but the analysis revealed the criteria that made the most significant contribution to the clustering algorithm, which allowed the rest to be excluded from consideration.

For the experiment, several driving cycles of different durations were generated from the processed and unprocessed data, and then the deviation of each of them from the actual record was calculated. This method of estimation is the most objective, as it shows both the percentage of deviation from real records and the dependence of the deviation on the cycle duration. Figure 8 shows the comparison result.

![Fig. 8. Dependence of the convergence of a cycle with a complete set of data on statistical parameters](image)

When analyzing the graphs in Figure 8, it can be seen that the cycles obtained from the raw data converge significantly worse than the real record. For example, cycles obtained from
processed data have a deviation of less than 5% for a duration of 10 hours, while cycles from raw data reach 5% only for a duration of 20 hours.

9 Conclusion

Based on the results of the work, it can be concluded that the processing of vehicle traffic statistics is an integral and extremely important part of working with statistical data, since it allows to get rid of data deficiencies caused by various physical aspects, such as a long period of sending a message to sensor equipment, inaccuracies in measurements, any features of measuring equipment, and so on. The paper also presents techniques for working with data to ensure high-quality operation of machine learning algorithms in relation to statistical data on vehicle movement and shows their effectiveness in generating driving cycles using the "micro-trip" method with clustering. As a measure of efficiency, statistical parameters were used to compare driving cycles. When comparing the quality of driving cycles generated from processed and unprocessed data, it turned out that cycles based on processed data provide better convergence with the original data set. The driving cycle obtained from the processed data provides a deviation of less than 5% from the actual traffic records, with half the duration compared to the raw data. Thus, in the case of the problem of generating driving cycles, the proposed solution to the problem of data quality allows to create a shorter driving cycle and reduce the time spent on conducting virtual tests of vehicles.

Acknowledgements

This research work was financially supported by the Ministry of Science and Higher Education of the Russian Federation under agreement No. 075-11-2022-015 dated 07.04.2022 with PJSC "KAMAZ" under the complex project "Creation of high-tech production of mechatronic transmissions for promising KAMAZ trucks and buses with electrical energy storage devices and hydrogen fuel cells" in terms of R&D and technological works.

References


5. S. V. Gusakov, V. A. Markov, D. V. Mikhryachev, Computational and experimental methodology for adjusting the driving cycle for the phase of vehicle movement in urban conditions, Proceedings of higher educational institutions, Moscow, Mechanical Engineering, EDN: OXDJPP, 5, 23-30 (2012)


9. D.A. Moiseikin, S.A. Kozhevnikova, Innovative processes in the transport industry, Concept, 2034(04), 14538


12. V. Vyugin, Mathematical foundations of machine learning and forecasting

13. Soledad Galli, A Cookbook on Developing Python functionality

14. Xinyi Jia, Huw Wang, Liangfei Xu, Qing Wang, Hang Li, Junyan Hu, Jianqiu Li, Mingao Ouyang, Constructing a representative driving cycle for a heavy-duty vehicle based on the Markov chain method taking into account road slope, energy and artificial intelligence, **6**, 100115 (2021)
