# Driver Identification with OBD-II Public Data

Kirill Uvarov
ETU “LETI”
St.Petersburg, Russia
uvarovkirill73@gmail.com

Andrew Ponomarev
SPC RAS
St.Petersburg, Russia
ponomarev@iias.spb.su

*Abstract*—**The problem of driver identification is an urgent one, and many publications propose various methods to solve it. Most of these papers use sensor data from the car's CAN bus to get a driver's fingerprint. However, at the moment, not all the sensors mentioned in these works are available without knowing the protocol details of a particular vehicle model, and this fact severely limits practical use of the existing methods in many applications. This paper explores the possibility of identifying the driver using information only from sensors listed in SAE J1979 specification. It turns out, that it is still possible to identify vehicle driver using these sensors. However, the identification accuracy decreases by approximately 15% compared to methods, in which the driver is identified with the full set of sensors, and is 79% versus 91% on the full set of sensors (on the dataset of ten drivers).**

## I. Introduction

A modern car is not just a vehicle, but a full-fledged smart device with various multimedia functions, additional security systems and a large number of different sensors. Until the 70th year of the last century, any car was equipped with a maximum of three sensors: fuel level, coolant temperature and oil pressure. They were connected to magnetoelectric and light display devices on the instrument panel. Their purpose was only to inform the driver about the features of the engine and the amount of fuel. At that time, the device of car sensors was very simple. Today, a car is a computer network consisting of at least 20 specific microcomputers in its composition and on average even with 30-40 network members. All these microcomputers communicate with each other in real time, so that the driver's trip is safe and comfortable, and the engine operation is efficient.

The number of connected cars is projected to reach 353 million by 2023. Connected cars are cars that communicate with other systems outside of the car. These systems can be other cars, various devices (traffic lights, garage doors), networks and services. Services can provide discounts on insurance services depending on the driving style [1], monitor the condition of the car in real time and advice on maintenance, route information, tips for improving driving style and fuel consumption. Connecting a car to third-party servers and collecting data from it poses security problems, and the car becomes more vulnerable to theft. As cars become more intelligent, attacks targeting them also become more intelligent. In modern times, attackers use various methods to steal a car key. It is enough for the hijacker to get into the car and start it, after that the car becomes at his/her full disposal. For additional protection, it is suggested to use biometric authentication tools based on both the driver's physiological characteristics, such as fingerprinting, scanning the eye shell, face or voice recognition, and behavioral characteristics. One of the ways to improve the security is to analyze the driver's driving style. Each driver has specific driving habits: specific acceleration and braking habits, overtaking maneuvers, and changing lanes – all of which can be used to identify the driver, and therefore, detect situations of unexpected driver’s change (due to the theft). The features of the driver's driving style are reflected in the trip's telematics data and can be deduced with a help of intelligent data analysis methods. In addition, driver identification can be an important source of personalization for various in-car infotainment systems and driver support applications.

Most of the data characterizing the driving style can be accessed via the vehicle's CAN bus. CAN (Controller Area Network) is an internal network that connects all the sensors of the car and through which they communicate with each other. It is very difficult to connect directly to it; therefore, the OBD (On-board Diagnostic) interface is typically used, allowing one to request data from the CAN bus. However, obtaining the required data sometimes can be problematic. Although a huge amount of data passes through the data bus every second, one needs to know the identifier (ID) of each specific parameter that needs to be extracted. Most parameter IDs are vendor-specific, non-public (not described in public documentation), therefore, it is very difficult to get these sensors' data (as well as interpret them). For example, such parameters as brake pedal position, steering angle and some others that are obviously very informative in identifying the driver are not public, therefore, their OBD IDs are not described in public documentation and may vary on different vehicle models. The authors of previous studies [2], [3], [4], that address the problem of driver identification using CAN bus data, use many non-public features, which allows to obtain good identification accuracy, but may severely limit the practical applicability of these methods (especially, with respect to applications, not associated with particular car vendor).

In this paper, we investigate driver identification using telematics data, obtained from the vehicle's CAN bus via OBD-II interface. Specifically, in this paper we analyze the role of non-public parameters in identifying the driver and evaluate the possibility of identifying the driver using only public ones. Therefore, unlike the earlier works that addressed this problem, we construct driver identification models using only public telematics data (with public OBD-II sensor IDs) and compare the quality of identifying multiple drivers with the quality of identifying multiple drivers based on a dataset including also non-public parameters. To do this, we review the results of past research (sec. II), consider the available dataset (sec. III), and possible ways of obtaining data from the car's CAN bus (sec. IV). We then summarize main machine learning techniques used for driver identification (sec. V), describe the selected features, data preprocessing pipeline (sec. VI), and the results of identifying drivers with the developed features (VII). Finally, in Section VIII, we discuss the results obtained and answer the question whether it is possible to identify the driver using only open CAN bus features.

## II. Related work

In this section, we discuss previous works that have addressed the problem of identifying the drivers for telematics data. The review highlights the methods that were used for the identification task, as well as the data used for analysis.

There are various ways to get telematics data about a trip. The authors of [5], [6] used data obtained in a car driving simulator. The researchers in [5] investigated driver behavior signals that are observed when a driver follows another car. The following signals were used for the study: use of the accelerator pedal, brake pedal, car speed and distance to the car in front. In the work [5] using the Gaussian Mixture Model (GMM) achieved 81% prediction accuracy with 12 drivers and 73% for 30 drivers. The authors of [6] investigated the features of the overtaking style for each driver. Using the accelerator and steering data, it was possible to achieve 85% accuracy of the prediction for 20 drivers. The Hidden Markov model (HMM) was used for the analysis.

Other researchers received telematics data using a smartphone. Sensors that are used in a smartphone: GPS, accelerometers, magnetometers, gyroscopes. Thanks to these sensors, researchers can get information about acceleration, speed, rotation speed, etc. In the works [7], [8], [9] this data is used for profiling drivers and other tasks. A study of the problem of driver identification using an inertial sensor describes in the work [10]. Using SVM and k-means methods, the authors managed to achieve 60% accuracy between 2 drivers.

One of the most reliable sources of information about the movement of a car is the car itself. As mentioned earlier, the car has a huge number of sensors that exchange telematics data with each other. To get real-time data, there is need to connect to the vehicle's internal network. In the works [2], [3], [11], [12], data obtained from the car's CAN bus using an OBD-II dongle is used. The authors of [12] using information from 5 car sensors using SVM, Random Forest, Naive Bayes, KNN methods were able to achieve 99% accuracy among 15 drivers. Kwak, Woo and Kim with a dataset from data from 51 sensors, got an accuracy of about 95% among 10 drivers [2].

Neural network and deep learning technologies have made a big step forward over the past few years. In recent published papers, some of the neural network methods were used to solve the problem of driver identification. The authors of [13] used convolutional neural network in their work and achieved 99% accuracy for 10 drivers using data from the car's CAN bus. In the article [4], LSTM-Recurrent Neural Network uses for driver identification. This solution not only showed high accuracy of driver identification, but also showed a significant advantage of his method before other methods when using data with noise.

From the review of works, we can highlight the point that this problem has been solved for a long time and it is relevant. Different algorithms and different data sources are used to solve the problem. The sources are mainly smartphone sensors and data from the car's CAN bus. The main algorithms that showed the greatest efficiency: random forest, SVM, KNN, neural networks.

## III. Available dataset

In order to identify the driver, there is a need to get telematics data of the trip. As part of the work, the data received from the CAN bus of the car via the OBD-II dongle will be considered. Looking at past work, the authors of [2], [3], [4] use the OCSLab security dataset for research [14]. The data was collected in South Korea, using a KIA Motors Corporation vehicle. The experiment involved 10 drivers who passed the route length of 23 km. There were 3 types of roads on the route: city road, freeway, and parking. Each type of road had its own characteristics. There were traffic lights and pedestrian crosswalks on the city road, but none on the motorway. In the parking lot, the driver had to drive slowly and carefully. For reliability, each driver drove the route back and forth. Data was collected via OBD-II and CarbigsP (OBD-II scanner).

It is difficult to get a dataset similar to the OCSLab security dataset or sensors used in the work [12]. There is a limited set of open ODB-II identifiers [15] and the most informative sensors, such as the angle of rotation of the wheel, pressing the brake pedal, etc. are unavailable. Sensors that are not included in the list of open sensors may have different IDs for each manufacturer and manufacturers do not put them in the public domain. Moreover, different models from the same manufacturer may also have different IDs. One of the options to get the necessary IDs is reverse engineering CAN bus, when the OBD-II dongle listens and records all the data that passes through the CAN bus and then this traffic is analyzed in special software.

It is not always possible to get information from the desired sensors. Therefore, in this work, we will consider the problem of identifying drivers using data only from sensors whose IDs in OBD-II are publicly available and anyone with an OBD-II dongle can get data from them.

## IV. CAN bus and OBD-II

As mentioned earlier, a modern car consists of a large number of microcomputers that control various car systems. These microcomputers are called Electronic Control Units (ECUs). The safety and comfort of trip depends on real-time data exchange between different ECUs. The ECU is responsible for the detection of accidents, the implementation of anti-lock braking, airbag control, etc. Typically, ECUs are networked on one or more buses based on the Controller Area Network (CAN) standard. ECUs communicate with each other using packets. The packet that is sent over the bus is visible to all blocks, and each network component decides whether the packet is intended for it.

Getting data directly is a rather difficult task, since the entire CAN bus is located inside the car. To access it, one needs to get into the wiring of the car, which may cause damage and the inability to continue using the car. A simpler method is OBD-II. OBD-II is an on-Board vehicle diagnostic standard that has been mandatory for cars sold in the United States since 1996 In Europe, the equivalent of OBD-II is EOBD. The OBD-II standard specifies a standardized hardware connector, its pinout, available protocols and message format. This connector is connected directly to the CAN bus, therefore, it can be used it to get the necessary data. To do this, one should send a message in a format understandable by OBD-II, which is converted to the format of the CAN packet, and passed to the bus. Data is received in reverse order. There are various OBD-II adapters that can be connected via wire/Bluetooth/Wi-Fi and used to obtain real-time telematics data.

Obtaining telematics data from OBD-II is regulated by SAE J1979 [16]. The ISO 15031 standard is based on this document. The SAE J1979 standard defines several diagnostic modes, such as: show current data, show freeze frame data, show saved diagnostic trouble codes, and others. To get data by sending the mode number and parameter ID (PID). Each parameter has its own PID. The ECU must respond to the message and send the requested data value. All data values returned for sensor readings will be actual readings, not default values or substitutes used by the system due to a malfunction of this sensor. The table of open PIDs and available operating modes is also specified in [16]. An illustration of the telematics data extraction from the vehicle is shown in Fig. 1.The main problem is that manufacturers are not required to implement all the PIDS listed in [16], and they are allowed to include their own PIDS that are not listed. For this reason, it is very difficult to collect datasets similar to those in section III from any vehicle. A short list of OBD-II parameters is shown in Table I.

TABLE I. SHORT LIST OF OBD-II PARAMETERS

| PID (hex) | Data bytes returned | Description | Min value | Max value | Units |
|---|---|---|---|---|---|
| 0C | 2 | Engine speed | 0 | 16,383.75 | RPM |
| 0D | 1 | Vehicle speed | 0 | 255 | km/h |
| 0F | 1 | Intake air temperature | -40 | 215 | °C |
| 49 | 1 | Accelerator pedal position D | 0 | 100 | % |
| 11 | 1 | Throttle position | 0 | 100 | % |
| 63 | 2 | Engine reference torque | 0 | 65,535 | Nm |

## V. DRIVER IDENTIFICATION METHODS

Based on the works [9], [10], [11], [12] we highlight several machine learning algorithms that have shown good results in identifying drivers.

### *A. KNN*

The nearest neighbor method is one of the simplest machine learning algorithms. Although it is quite simple, it is a fairly powerful algorithm that provides fairly high accuracy for most tasks [17]. The KNN method uses the well-known principle of «*Cicero pares cum paribus facillime congregantur*» («Birds of a feather flock together») and tries to classify an unknown sample based on the known classification of its neighbors. An important parameter of this method is $k$ – the number of nearest neighbors, based on which the driver class will be determined. We analyze the accuracy of the method with different values of $k$. The Euclidean distance will be used as the distance measure.

### *B. Decision tree*

A decision tree is a hierarchical tree structure that displays decision points that are formed from rules of the form " If ..., then ...". Rules are automatically generated during training on the training set. Analytical models in the form of decision trees are more verbalizable, interpretable, and understandable to humans. Another advantage of this type of algorithm is the relatively low cost of data preparation. As hyperparameters, we will consider the maximum depth of the tree. The deeper the tree, the more divisions it has and the more information it has about the data.

### *C. Random forest*

The forest consists of a large number of individual decision trees that act as an ensemble. The forecast is obtained by aggregating responses from multiple trees. The fundamental idea of this algorithm is obvious and effective – the wisdom of the crowd. A large number of relatively uncorrelated trees will surpass any single tree. An important hyperparameters that we consider are the number of trees and the maximum depth of each tree.

### *D. Gradient boosting*

Gradient boosting is a machine learning technique for classification and regression problems that builds a prediction model in the form of an ensemble of weak predictive models, usually decision trees. The main idea of boosting is to combine weak functions that are built during an iterative process, where at each step a new model is trained using error data from previous ones. A hyperparameters that we consider are the number of trees, learning rate and tree depth.

### *E. Support Vector Machine*

SVM is used for solving the binary classification problem. The main idea of this algorithm is to find the best line or boundary to divide an n-dimensional space into classes. This separation makes it easy to put a new data point in the correct category when solving a classification problem. The boundary of the best solution is called a hyperplane. There are several drivers in the dataset that is used for identification research. The other algorithms solve multiclass classification problems. SVM solves the binary classification problem and will treat the problem as one against all. If we consider the problem as a method of driver authentication, when it is necessary to decide whether the driver is correct or not, then this algorithm should cope with this task.

## VI. METHODOLOGY

### *A. Data description*

We use the modified OCS Lab security dataset as the data for the study [14]. All data from sensors whose IDs are not specified in SAE J1979 have been removed from it. We also

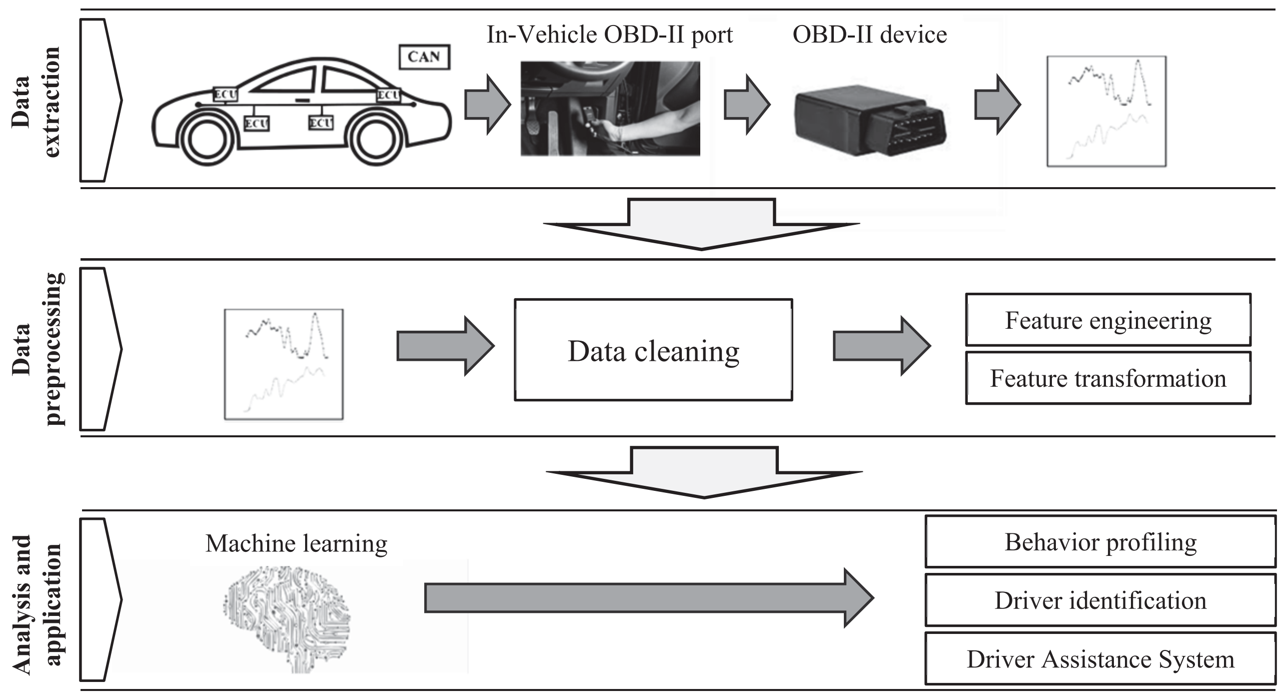

Fig. 1. Telematics data extraction from the vehicle

have removed strongly correlated features such as *Absolute_throttle_position* and *Throttle_position_Signal*, and the data from the Short Term Fuel Bank (STFB) sensor. Instead, the Long term fuel bank is taken into account, which is based on the STFB values. The list of features that will be used for the study is presented in Table II.

*B. Feature engineering*

In several previous works [18], [19], additional features such as Acceleration Vehicle Speed (1) and Jerk Vehicle Speed (2) were calculated based on the data obtained. We will also add these features to the main dataset. These features can show us sudden changes in speed, which may be the driver's personal characteristic. Acceleration was computed using the following equation:

$$a_x(t_2) = \frac{v_x(t_2) - v_x(t_1)}{t_2 - t_1} \tag{1}$$

where $t_2$ is the current time stamp, and $t_1$ is the previous time stamp.

Successively the jerk vehicle speed was computed as:

$$j_x(t_2) = \frac{a_x(t_2) - a_x(t_1)}{t_2 - t_1} \tag{2}$$

*C. Sliding window*

After adding additional features all trips were divided into moving windows. The length of the window and its pitch are selected during the experiment. The following statistical features were calculated for each window: mean, median, variance, standard deviation, maximum, skewness and kurtosis. Thus, the number statistical features were obtained is 84.

*D. Normalization*

Data from sensors have different measurement units and scales, so we normalized data using Min-Max normalization. This transformation is mandatory for the KNN and SVM algorithms. The data were normalized according to the following formula:

$$X_i = \frac{(x_i - \min(x_i))}{\max(x_i) - \min(x_i)}, \tag{3}$$

where min is the minimum value of feature data, and max is the maximum value of the feature respectively.

*E. Evaluation Metrics*

The task of identification is to correctly identify the driver. Assuming that each driver can be viewed as a separate class, the identification problem can reduced either to binary classification (when we need to determine the correct driver or the wrong one), or to the multi-class identification problem (when we need to determine the particular driver). Therefore, typical classification metrics used for classification were selected as metrics for evaluating the results of driver identification.

Accuracy is the most frequently used quality metric for identifying drivers in previous works. This metric will also be calculated in this paper. This calculation will be performed to compare changes in this metric for different features, such as features that are publicly available via OBD and all features that are available in this data set. Additionally, 2 more metrics will be calculated: precision and f-score.

No doubt, that in practical applications other criteria for evaluation of driver identification methods can also be important (for example, computational or memory complexity, achieving certain ML model learning characteristics in limited conditions). However, the main goal of this work is not to propose a new method of driver identification, instead, the goal is to assess the impact of non-public features on the accuracy. Therefore, we use well-known methods, including those that have performed well in similar works.

## VII. Results and discussions

Cross-validation for 5 folds is used to evaluate model generalization. The model is trained using data of 4 blocks, and is evaluated on the data of the 5th. Before cross-validation, data is not shuffled, so that it does not happen that the train set contains data in the ith and (i + 2)th time window, and the validation set contains the (i+1)th time window.

TABLE II. The selected features for the study

| Feature | Description | Range | Units |
|---|---|---|---|
| 'Accelerator_Pedal_value | The accelerator pedal sensor transmits the position of the accelerator pedal to the engine control unit. | [0;100] | % |
| 'Intake_air_pressure' | The measured value of the intake manifold pressure sensor are required to calculate the intake air mass. | [0;255] | kPa |
| 'Absolute_throttle_position' | Actual position of the throttle | [0;100] | % |
| 'Long_Term_Fuel_Trim_Bank1' | Fuel adjustment as a percentage in the long term | [-100;99.2] | % |
| 'Engine_speed' | Number of revolutions per minute performed by the engine | [0;16383.75] | RPM |
| 'Torque_of_friction' | Friction torque is the torque caused by the frictional force that occurs when two objects in contact move. | [-125;130] | % |
| 'Engine_coolant_temperature | Temperature of the coolant/antifreeze mix in the cooling system, giving an indication of how much heat the engine is giving off | [-40;215] | C |
| 'Engine_torque' | It is a rotating force produced by an engine's crankshaft. | [-125;130] | % |
| 'Calculated_LOAD_value' | This value indicates a percentage of peak available torque. | [0;100] | % |
| 'Vehicle_speed' | 2 It is current speed of the vehicle | [0;255] | km/h |

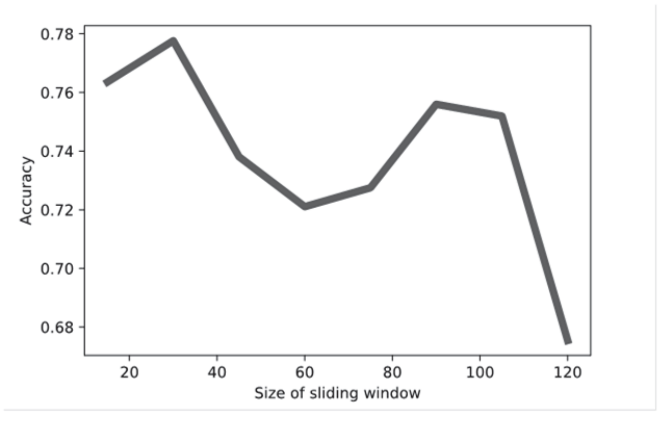

Fig. 2. Accuracy depends on the size of the sliding window

We have analyzed the effect of the size of a moving window on the accuracy of driver prediction. The sliding window should capture the driver's fingerprint. If the window size is small, then the features of each driver will be impossible to catch. If the window size is large, the driver's fingerprint may be blurred due to the possible large number of events that occurred on the road during this time period. The Fig. 2 shows the dependence of accuracy on the window size. A similar dependence of accuracy on the window size was found in [4].

The window size was selected as a result of studying the dependence of accuracy on the size of the time window, which is shown in Fig. 2. The size of the sliding window is 30 seconds in increments of 1 second.

For each classification algorithm, section V discussed hyperparameters that will be further selected for the best accuracy of the algorithm. For the random forest algorithm, the following features are configured: the number of trees and the depth of each tree. By selecting the value of hyperparameters, accuracy was improved to 0.74 compared to 0.79 for the default features. The value of the selected features: number of trees – 900, maximum depth – 30. For the decision tree algorithm, selecting the maximum depth parameter did not significantly change accuracy: 0.68 for the standard value and 0.72 for the selected value of 63. For the KNN algorithm, selecting the «number of neighbors» parameter did not change accuracy and the value remained at 0.5. It was able to improve the accuracy from 0.65 to 0.78 for the gradient boosting algorithm. Hyperparameter values are: number of trees – 1000, learning rate – 0.03, a tree depth – 9.

To solve the classification problem, we used 5 algorithms. For each algorithm, we calculated quality metrics for full OCSlab security dataset, each fold and on average. The results for the random forest algorithm are shown in Table III, for the Decision tree algorithm in Table IV, for the KNN algorithm in Table V, for Gradient boosting algorithm in Table VI and for SVM algorithm in Table VII.

For comparison, quality metrics were first calculated on the full dataset. Accuracy on the random forest algorithm was 91%, which generally coincides with the accuracy in other works that study the full OCSLab security dataset. Similarly, we calculated accuracy on a dataset with open parameters and got a value of 79%. The decrease in accuracy can be explained by the fact that we removed some informative parameters that the researchers used from the full dataset. The most important features in the random forest algorithm are shown in the Fig. 3.

The accuracy of guessing drivers depends on their total number. The maximum accuracy will be when you need to guess from 2 drivers, when the identification problem is a binary classification problem. As the number of drivers increases, the accuracy of guesses will decrease. In order to study how the accuracy changes when increasing the number of drivers we formed combinations of only 2, 3, etc. drivers and calculated the accuracy of detecting a driver in these combinations. Fig. 4 shows the results.

If we randomly guess the driver, the theoretical accuracy will be 10%. In our results, we obtained an accuracy almost 80%, which can be considered a good result. If we use driver identification as a means of protecting the car from theft, then the problem becomes a binary classification, when we need to decide whether the driver is right or wrong. With this problem, the result of our research showed that we can identify the driver with 99% accuracy.

TABLE III. THE RESULTS FOR RANDOM FOREST ALGORITHM

| | **Full dataset** | **Dataset with open PIDs** | | | | | |
|---|---|---|---|---|---|---|---|
| | | 1 Fold | 2 Fold | 3 Fold | 4 Fold | 5 Fold | Mean |
| Accuracy | 0.91 | 0.7 | 0.84 | 0.81 | 0.84 | 0.74 | 0.79 |
| Precision | 0.88 | 0.65 | 0.82 | 0.79 | 0.81 | 0.8 | 0.77 |
| F-Score | 0.81 | 0.62 | 0.79 | 0.77 | 0.82 | 0.71 | 0.74 |

TABLE IV. THE RESULTS FOR DECISION TREE ALGORITHM

| | **Full dataset** | **Dataset with open PIDs** | | | | | |
|---|---|---|---|---|---|---|---|
| | | 1 Fold | 2 Fold | 3 Fold | 4 Fold | 5 Fold | Mean |
| Accuracy | 0.74 | 0.56 | 0.8 | 0.78 | 0.77 | 0.68 | 0.72 |
| Precision | 0.73 | 0.57 | 0.79 | 0.72 | 0.7 | 0.72 | 0.7 |
| F-Score | 0.69 | 0.53 | 0.78 | 0.7 | 0.71 | 0.63 | 0.67 |

TABLE V. THE RESULTS FOR KNN ALGORITHM

| | **Full dataset** | **Dataset with open PIDs** | | | | | |
|---|---|---|---|---|---|---|---|
| | | 1 Fold | 2 Fold | 3 Fold | 4 Fold | 5 Fold | Mean |
| Accuracy | 0.64 | 0.47 | 0.52 | 0.53 | 0.51 | 0.46 | 0.5 |
| Precision | 0.62 | 0.46 | 0.51 | 0.5 | 0.49 | 0.45 | 0.48 |
| F-Score | 0.59 | 0.43 | 0.48 | 0.49 | 0.48 | 0.42 | 0.46 |

TABLE VI. THE RESULTS FOR GRADIENT BOOSTING ALGORITHM

| | **Full dataset** | **Dataset with open PIDs** | | | | | |
|---|---|---|---|---|---|---|---|
| | | 1 Fold | 2 Fold | 3 Fold | 4 Fold | 5 Fold | Mean |
| Accuracy | 0.91 | 0.67 | 0.83 | 0.81 | 0.84 | 0.75 | 0.78 |
| Precision | 0.91 | 0.67 | 0.82 | 0.78 | 0.84 | 0.8 | 0.78 |
| F-Score | 0.85 | 0.62 | 0.79 | 0.78 | 0.84 | 0.7 | 0.75 |

TABLE VII. THE RESULTS FOR SVM ALGORITHM

| | **Full dataset** | **Dataset with open PIDs** | | | | | |
|---|---|---|---|---|---|---|---|
| | | 1 Fold | 2 Fold | 3 Fold | 4 Fold | 5 Fold | Mean |
| Accuracy | 0.78 | 0.58 | 0.66 | 0.63 | 0.72 | 0.64 | 0.65 |
| Precision | 0.78 | 0.54 | 0.65 | 0.65 | 0.68 | 0.64 | 0.63 |
| F-Score | 0.7 | 0.49 | 0.62 | 0.61 | 0.67 | 0.6 | 0.6 |

## VIII. CONCLUSION AND FUTURE WORK

The main goal of this study is to solve the problem of driver identification using telematics data. This work has an important difference from other studies. Telematics data that was obtained using the car's OBD-II was also used in previous works. But most of the parameters that the authors captured are difficult to get, because the IDs of these parameters are non-public. In this work, we used only public OBD-II identifiers. Several machine learning algorithms were used to solve the identification problem. Quality metrics were obtained for each algorithm, both for a dataset with the entire set of parameters, and for a dataset with only public parameters. The highest value was achieved using the random forest algorithm, which turned out to have the accuracy value of 0.79 for 10 drivers (for the full dataset the accuracy is 0.91). In general, for each algorithm, the accuracy decreased approximately by 15%.

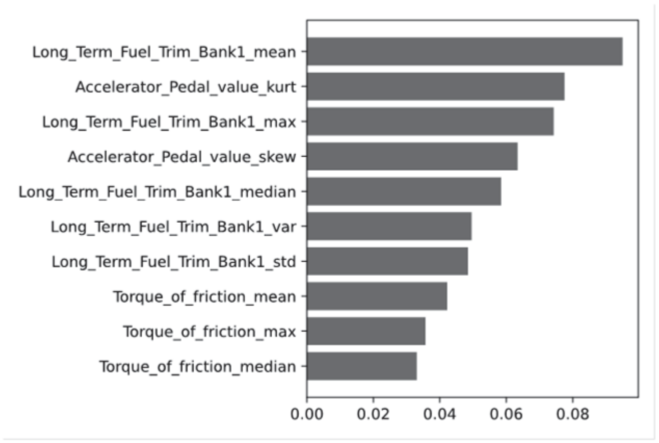

Fig. 3. Random forest feature importance

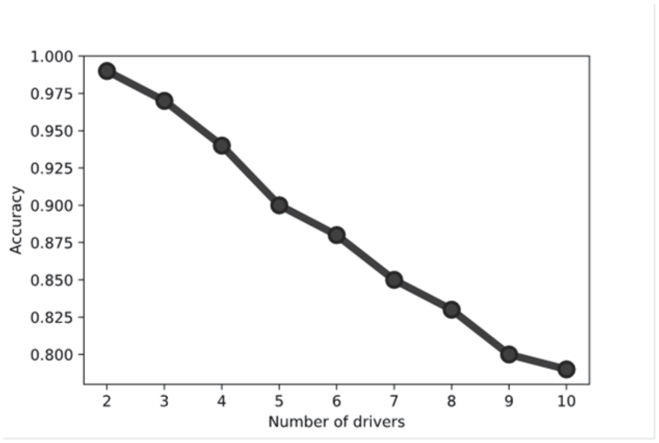

Fig. 4. Random forest accuracy depending on the number of drivers

This decrease is explained by the absence of some informative parameters in the dataset with open parameters. If identification is used as an anti-theft method, the accuracy of guessing the real owner is 99%. As the number of drivers increases, the difficulty of the task increases and the accuracy of identification decreases. To further increase the identification accuracy it is possible to leverage other sources of information (beside OBD-II), for example, place accelerometer in the vehicle (as a separate device or use smartphone). It is also possible to capture additional open OBD-II parameters and add various statistical features and their interactions.

In this work, we used the open parameters of OBD-II, present in the OCSLab security dataset. However, this dataset has a limited set of open OBD-II parameters; the whole list of public OBD-II parameters is longer. In the future, we will explore other public parameters of OBD-II and evaluate their usefulness for the driver identification problem. It will allow to make a list of the most informative for driver identification parameters available for application developers not affiliated with car vendors. Further, the identified parameters will be used in the driver support system being developed, where driver identification is one of the core features.

## ACKNOWLEDGMENT

The study was supported by Russian State Research, project 0073-2019-0005.

## REFERENCES

[1] M.F. Carfora, F. Martinelli, F. Mercaldo, V. Nardone, A. Orlando, A. Santone and G.Vaglini, "A "pay-how-you-drive" car insurance approach through cluster analysis", *Soft Computing,* vol.23, 2019, pp. 2863–2875.

[2] B.I. Kwak, J. Woo and H.K. Kim, "Know your master: Driver profiling-based anti-theft method", *14th Annual Conference on Privacy, Security and Trust (PST) 2016*, Dec.2016, pp. 211-218.

[3] S. Ullah and D. Kim, "Lightweight Driver Behavior Identification Model with Sparse Learning on In-Vehicle CAN-BUS Sensor Data", *Sensors*, vol.20(18), 2020, 5030.

[4] A. Girma, X. Yan and A. Homaifar, "Driver Identification Based on Vehicle Telematics Data using LSTM-Recurrent Neural Network", *IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, 2019, pp. 894-902.

[5] T. Wakita, K. Ozawa, C. Miyajima, K. Igarashi, I. Katunobu, K. Takeda and F. Itakura, "Driver identification using driving behavior signals", *IEICE TRANSACTIONS on Information and Systems*, vol.89(3), 2006, pp.1188–1194.

[6] X. Zhang, X. Zhao, and J. Rong, "A study of individual characteristics of driving behavior based on hidden markov model", *Sensors & Transducers*, vol.167(3), 2014, pp.1726-5479.

[7] G. Castignani, T. Derrmann, R. Frank and T. Engel, "Driver behavior profiling using smartphones: A low-cost platform for driver monitoring", *IEEE Intelligent Transportation Systems Magazine*, vol.7, 2015, pp.91–102.

[8] A. Kashevnik, I. Lashkov and A. Gurtov, "Methodology and mobile application for driver behavior analysis and accident prevention", *IEEE Transactions on Intelligent Transportation Systems*, vol.21, 2020, pp.2427–2436.

[9] J. Warren, J. Lipkowitz and V. Sokolov, "Clusters of driving behavior from observational smartphone data", *IEEE Intelligent Transportation Systems Magazine*, vol.11, 2019, pp.171–180.

[10] M. V. Ly, S. Martin and M.M. Trivedi, "Driver classification and driving style recognition using inertial sensors", *IEEE Intelligent Vehicles Symposium (IV),* 2013, pp.1040-1045.

[11] S. Choi, J. Kim, D. Kwak, P. Angkititrakul and J. H. Hansen, "Analysis and classification of driver behavior using in-vehicle CAN-BUS information", *Biennial Workshop on DSP for In-Vehicle and Mobile Systems*, 2007, pp. 17–19.

[12] M. Enev, A. Takakuwa, K. Koscher and T. Kohno, "Automobile driver fingerprinting", Proceedings on Privacy Enhancing Technologies, vol.1, 2016, pp.4-19.

[13] Y. Xun, J. Liu, N. Kato, Y. Fang and Y. Zhang, "Automobile Driver Fingerprinting: A New Machine Learning Based Authentication Scheme",*IEEE Transactions on Industrial Informatics*, vol.16, 2020, pp.1417-1426.

[14] Hacking and Countermeasure Research Lab official website, Driving dataset, Web: https://ocslab.hksecurity.net/Datasets/driving-dataset.

[15] OBD-II PIDs, Web: https://en.wikipedia.org/wiki/OBD-II_PIDs.

[16] Society of Automotive Engineers official website, SAE J1979, Web: https://www.sae.org/standards/content/j1979_201702.

[17] D. Hand, H. Mannila and P. Smyth, *Principles of Data Mining*, The MIT Press, 2001, 546 pages.

[18] H. Alhamdan and M. Jilani, "Machine Learning for Automobile Driver Identification Using Telematics Data", *Advances in Data Science, Cyber Security and IT Applications*, 2019

[19] B.Wang, S. Panigrahi, M. Narsude and A. Mohanty, "Driver identification using vehicle telematics data", *SAE World Congress Experience*, 2017