Who's Driving My Car? A machine learning based approach to driver identification

Fabio Martinelli[1], Francesco Mercaldo[1], Vittoria Nardone[2], Albina Orlando[3], Antonella Santone[4]
[1]*Istituto di Informatica e Telematica, Consiglio Nazionale delle Ricerche (CNR), Pisa, Italy*
[2]*Department of Engineering, University of Sannio, Benevento, Italy*
[3]*Istituto per le Applicazioni del Calcolo "M. Picone", Consiglio Nazionale delle Ricerche (CNR), Napoli, Italy*
[4]*Department of Bioscience and Territory, University of Molise, Pesche (IS), Italy*
{*fabio.martinelli, francesco.mercaldo*}*@iit.cnr.it, vnardone@unisannio.it,*
a.orlando@iac.cnr.it, antonella.santone@unimol.it

Keywords: Car, can, obd, authentication, machine learning, identification.

Abstract: Despite the development of new technologies, in order to prevent the stealing of cars, the number of car thefts is sharply increasing. With the advent of electronics, new ways to steal cars were found. To avoid auto-theft attacks, in this paper we propose a machine leaning based method to silently e continuously profile the driver by analyzing built-in vehicle sensors. We evaluate the efficiency of the proposed method in driver identification using 10 different drivers. Results are promising, as a matter of fact we obtain a high precision and a recall evaluating a dataset containing data extracted from real vehicle.

1 INTRODUCTION AND BACKGROUND

As highlighted by several studies, car theft is increasing around the globe and the phenomenon does not appear to stop.

The FBI national crime statistics show that car theft appears to be on the rise this year in the United States. While burglary and larceny theft were down by 10% and 3% respectively, car theft was up by 1%[1].

As a matter of fact, the National Insurance Crime Bureau (NICB) reports that the Honda Accord was the most frequently stolen passenger vehicle in 2016, with 50,427 thefts among all model years of this car, while among 2016 model year vehicles, the Toyota Camry was the most frequently stolen vehicle in calendar year 2016, with 1,113 thefts, followed by the Nissan Altima with 1,063 thefts[2].

In last years, cars are equipped with many computers on board, exposing them to a new type of attacks (Martinelli et al., 2017; Alheeti et al., 2015; Lyamin et al., 2014). As a matter of fact, operating systems running on cars, like any other one, are exposed to bug and vulnerabilities (Taylor et al., 2016).

This scenario calls for a plethora of new car theft possibility (Massaro et al., 2017).

For instance, keyless cars could be at risk from attackers using simply radio transmitters with the aim to steal vehicles: in order to make evidence of this issue, the ADAC German company used radio transmitters to evaluate which cars could be broken in to. The BMW, Audi, Ford, Land Rover, Hyundai Renault and VW brands were among the manufacturers whose cars are at risk from hackers[3].

The main used technique consists in breaking into the vehicle and plugging a laptop into the hidden diagnostic socket used by garages to detect and solve faults: once connected the thieves can access the vehicle's electronic information, allowing them to drive it away.

Since cars are evolved with on-board computers, other developed techniques consist in get owners to install malicious software into their mobile devices working as a door lock in order to make the door open.

Researchers in last years proposed several approaches in order to solve the diver identification is-

[1] http://www.tracknstop.com/car-theft-in-u-s-on-rise-in-2016/

[2] http://www.iii.org/fact-statistic/auto-theft

[3] http://www.express.co.uk/life-style/cars/806889/Keyless-entry-car-keys-hack-theft-warning

sue. For instance, authors in (Wakita et al., 2006) propose a driver identification method that is based on the driving behavior signals that are observed while the driver is following another vehicle. They analyze signals, as accelerator pedal, brake pedal, vehicle velocity, and distance from the vehicle in front, were measured using a driving simulator. The identification rates were 81% for twelve drivers using a driving simulator and 73% for thirty drivers.

Researchers in (Miyajima et al., 2007; Nishiwaki et al., 2007) model gas and brake pedal operation patterns with Gaussian mixture model (GMM). They achieve an identification rate of 89.6% for a driving simulator and 76.8% for a field test with 276 drivers, resulting in 61% and 55% error reduction, respectively, over a driver model based on raw pedal operation signals without spectral analysis.

Driver behavior is described and modeled in (Choi et al., 2007) using data from steering wheel angle, brake status, acceleration status, and vehicle speed through Hidden Markov Models (HMMs) and GMMs employed to capture the sequence of driving characteristics acquired from the CAN bus information. They obtain 69% accuracy for action classification, and 25% accuracy for driver identification.

In reference (Meng et al., 2006) features extracted from the accelerator and brake pedal pressure are considered as inputs to a fuzzy neural network (FNN) system to ascertain the identity of the driver. Two fuzzy neural networks, namely, the evolving fuzzy neural network (EFuNN) and the adaptive network-based fuzzy inference system (ANFIS), are used to demonstrate the viability of the two proposed feature extraction techniques.

Starting from these considerations, in this paper we propose a method to detect car theft using machine learning techniques.

We highlight that our proposed method is able to reach a precision and a recall equal to 0.992, while the cited works obtain a detection rate lower than the one we reached.

Our method permits to define the driver profile by merging together information about his behavior, for this reason it can be useful to discriminate between car owner and impostors.

Using well-known machine learning algorithms, we classify the features set obtained from real-world cars employed in a real environment in order to evaluate the effectiveness of the features extracted.

The paper poses the following research question:

- *is it possible to characterize the driver behavior through a set features generated by himself/herself when he/she is driving?*

Below we highlight the main advantages provided by our method:

- the features can be captured by using the car built-in sensors without additional hardware;
- the features can be gathered with a good degree of precision and are not influenced by external factors (for instance noises, air impurity);
- the features can be collected while the user is driving the car: the driver is not required to enter any image or voice (this is the reason why the method is called silent).

The paper proceeds as follows: Section 2 deeply describes and motivates the detection method; Section 3 illustrates the results of experiments; finally, conclusions are drawn in Section 4.

2 THE METHOD

In this section we describe our method to identify drivers behavior using data retrieved by CAN bus.

These data are broadcast to all components on the bus and each component decides whether it is intended for them, although segmented CAN networks do exist.

In practice, using the CAN protocol the ECU “A” is able to send data to the ECU “B”, but this is not enough to realize the communication: it is also necessary that the ECU “B” is able to recognize and use the data received by ECU “A”, for this reason it is necessary something that make able the two electronic control units to “speak the same language. This is the reason why the OBD-II standard (On Board Diagnostics) (Birnbaum and Truglia, 2001) was introduced, in order to define a common language that make the various ECUs able to communicate (of Automotive Engineers, 1999).

We consider the full set of real data (Kwak et al., 2016) processed from the in-vehicle CAN data: to collect data, the On Board Diagnostics 2 (OBD-II) and CarbigsP as OBD-II scanner were used. The recent vehicle has many measurement sensors and control sensors, so the vehicle is managed by ECU in it. ECU is the device that controls parts of the vehicle such as Engine, Automatic Transmission, and Antilock Braking System (i.e., the ABS). OBD refers to the self-diagnostic and reporting capability by monitoring vehicle system in terms of ECU measurement and vehicle failure. The data are recorded every 1 second during driving and they are related to a recent model of KIA Motors Corporation in South Korea.

Ten different drivers participated to the experiment by driving, with the same car, 4 different round-trip path in Seoul (i.e., between Korea University and

SANGAM World Cup Stadium) for about 23 hours of total driving time.

The driving path consists of three types of city way, motor way and parking space with the total length of about 46 km. The experiment is performed since July, 28, 2015. The experiments was performed in the similar time zone from 8 p.m. to 11 p.m. on weekdays. The ten drivers completed two round trips for reliable classification, while data are collected from totally different road conditions. The city way has signal lamps and crosswalks, but the motor way has none. The parking space is required to drive slowly and cautiously.

The data that we used has total 94,401 records stored every second with the size of 16.7Mb in total and it is freely available for research purpose[4].

We designed an experiment in order to evaluate the effectiveness of the feature vector gathered through the CAN bus we propose, expressed through the research question RQ stated in the introduction.

More specifically, the experiment is aimed at verifying whether the features set is able to discriminate the car owner by impostors.

We consider the classification analysis in order to assess whether the feature set is able to correctly classify car owner and impostors.

We adopt the supervised learning approach, considering that the driver features evaluated in this work contain the driver labels.

Following classification algorithms were used: J48, J48graft, J48consolidated, RandomTree and RepTree. These algorithms were applied to all features (i.e., to the full feature vector gathered from the OBD).

The classification analysis was accomplished with Weka[5], a suite of machine learning software, largely employed in data mining for scientific research.

3 THE EVALUATION

Five metrics were used to evaluate the classification results: FP rate, Precision, Recall, F-Measure and ROC Area.

The false positive rate is calculated as the ratio between the number of negative driver traces wrongly categorized as belonging to the owner (i.e.,the false positives) and the total number of actual impostor traces (i.e., the true negatives):

$$FP\ rate = \frac{fp}{fp+tn}$$

where *fp* indicates the number of false positives and *tn* the number of true negatives.

The precision has been computed as the proportion of the examples that truly belong to class X among all those which were assigned to the class. It is the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved:

$$Precision = \frac{tp}{tp+fp}$$

where *tp* indicates the number of true positives and *fp* indicates the number of false positives.

The recall has been computed as the proportion of examples that were assigned to class X, among all the examples that truly belong to the class, i.e., how much part of the class was captured. It is the ratio of the number of relevant records retrieved to the total number of relevant records:

$$Recall = \frac{tp}{tp+fn}$$

where *tp* indicates the number of true positives and *fn* indicates the number of false negatives.

The F-Measure is a measure of a test's accuracy. This score can be interpreted as a weighted average of the precision and recall:

$$F\text{-}Measure = 2 * \frac{Precision * Recall}{Precision + Recall}$$

The Roc Area is defined as the probability that a positive instance randomly chosen is classified above a negative randomly chosen.

The classification analysis consisted of building classifiers in order to evaluate features accuracy to distinguish the car owner by an impostor.

We consider two different approaches in order to build the model starting from the features.

In the first one, the multi driver classification, for training the first classifier, we defined T as a set of labeled behavioral traces *(BT, l)*, where each BT is associated to a label $l \in \{A, B, C, D, E, F, G, H, I, J\}$.

For training the second classifier, i.e., the binary one, we defined T as a set of labeled behavioral traces *(BT, l)*, where each BT is associated to a label $l \in \{impostor, owner\}$. For each BT we built a feature vector $F \in R_y$, where y is the number of the features used in training phase (y=*51*).

For the learning phase, we consider a k-fold cross-validation (Mitchell, 1999; Refaeilzadeh et al., 2009): the dataset is randomly partitioned into k subsets. A single subset is retained as the validation dataset for

[4] https://sites.google.com/a/hksecurity.net/ocslab/Datasets/driving-dataset

[5] http://www.cs.waikato.ac.nz/ml/weka/

testing the model, while the remaining *k-1* subsets of the original dataset are used as training data. We repeated the process for $k = 10$ times; each one of the k subsets has been used once as the validation dataset. To obtain a single estimate, we computed the average of the k results from the folds.

We evaluated the effectiveness of the classification method with the following procedure:

1. build a training set $T \subset D$;
2. build a testing set $T' = D \div T$;
3. run the training phase on T;
4. apply the learned classifier to each element of T'.

Each classification was performed using 20% of the dataset as training dataset and 80% as testing dataset employing the full feature set.

We defined C_u as the set of the classifications we performed, where u identifies the driver ($1 \leq u \leq 10$).

For sake of clarity, we explain with an example the method we adopted in the binary classification: when we perform C_2 classification, we label the traces related to the driver #2 as owner traces, and the traces of the other user as impostor, while in the multi driver classification we consider the ten different label drivers.

The results that we obtained with this procedure are shown in Table 1.

In the multi driver classification (**All drivers** family) we obtain the following best results from the point of the views of the metrics we considered:

- FP rate equal to 0.001 with the J48, J48graft and J48consolidated algorithms;
- Precision, Recall and F-Measure equal to 0.992 using the J48 and the J48graft classification algorithms;
- Roc Area equal to 0.998 using J48, J48graft, J48consolidated and RepTree classification algorithms.

In the single driver classification, we obtain the following results:

- FPRate ranging from 0 to 0.0018;
- Precision ranging from 0.844 to 0.998;
- Recall ranging between 0.88 and 0.998;
- F-Measure ranging between 0.88 and 0.998;;
- Roc Area ranging between 0.911 and 1.

4 CONCLUSIONS AND FUTURE WORK

Modern vehicles, differently by older ones, integrate a lot of sophisticated electronic devices. This increasing technologies permitted to find new way to steal cars, for instance by exploiting the vulnerabilities of the operating system embedded in today's car. This scenario calls for new methodologies in order to stem the phenomenon resulting from the introduction of computers in the car, with the consequent vulnerability of software used. As a matter of fact, attackers are developing several way in order to steal vehicles exploiting the increasing technology currently available in nowadays cars. In this paper we propose a method able to discriminate an impostor by the car owner using a set of characteristics available by the sensor embedded into the car. Using machine learning techniques (Cimitile et al., 2017; Mercaldo et al., 2017; Mercaldo et al., 2016), we design several classifiers able to evaluate the effectiveness of our method: as a matter of fact we obtain in average a precision and a recall equal to 0.998 in car owner discrimination. As future work, we plan to take into account in our model the type of road in order to design a system able to advise the user about the driving style to adopt. In addition, we will extend the evaluation considering features extracted by trucks and motorcycles with the aim to identify thefts not only cars-related. Finally, another very interesting research direction involves the use of heuristic model checking (De Francesco et al., 2016). This is supported by the very promising results obtained in other fields, like for example in the malware detection area (Battista et al., 2016).

Acknowledgments

This work has been partially supported by H2020 EU-funded projects NeCS and C3ISP and EIT-Digital Project HII and PRIN "Governing Adaptive and Unplanned Systems of Systems" and the EU project CyberSure 734815.

REFERENCES

Alheeti, K. M. A., Gruebler, A., and McDonald-Maier, K. D. (2015). An intrusion detection system against malicious attacks on the communication network of driverless cars. In *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*, pages 916–921. IEEE.

Battista, P., Mercaldo, F., Nardone, V., Santone, A., and Visaggio, C. A. (2016). Identification of android malware families with model checking. In *Proceedings of the 2nd International Conference on Information Systems Security and Privacy, ICISSP 2016, Rome, Italy, February 19-21, 2016.*, pages 542–547. SciTePress.

Birnbaum, R. and Truglia, J. (2001). *Getting to know OBD II*. R. Birnbaum.

Choi, S., Kim, J., Kwak, D., Angkititrakul, P., and Hansen, J. H. (2007). Analysis and classification of driver behavior using in-vehicle can-bus information. In *Biennial Workshop on DSP for In-Vehicle and Mobile Systems*, pages 17–19.

Cimitile, A., Martinelli, F., and Mercaldo, F. (2017). Machine learning meets ios malware: Identifying malicious applications on apple environment. In *ICISSP*, pages 487–492.

De Francesco, N., Lettieri, G., Santone, A., and Vaglini, G. (2016). Heuristic search for equivalence checking. *Software and System Modeling*, 15(2):513–530.

Kwak, B. I., Woo, J., and Kim, H. K. (2016). Know your master: Driver profiling-based anti-theft method. In *PST 2016*.

Lyamin, N., Vinel, A. V., Jonsson, M., and Loo, J. (2014). Real-time detection of denial-of-service attacks in ieee 802.11 p vehicular networks. *IEEE Communications letters*, 18(1):110–113.

Martinelli, F., Mercaldo, F., Nardone, V., and Santone, A. (2017). Car hacking identification through fuzzy logic algorithms. In *Fuzzy Systems (FUZZ-IEEE), IEEE International Conference on. IEEE*. IEEE.

Massaro, E., Ahn, C., Ratti, C., Santi, P., Stahlmann, R., Lamprecht, A., Roehder, M., and Huber, M. (2017). The car as an ambient sensing platform. *Proceedings of the IEEE*, 105(1):3–7.

Meng, X., Lee, K. K., and Xu, Y. (2006). Human driving behavior recognition based on hidden markov models. In *Robotics and Biomimetics, 2006. ROBIO'06. IEEE International Conference on*, pages 274–279. IEEE.

Mercaldo, F., Nardone, V., and Santone, A. (2017). Diabetes mellitus affected patients classification and diagnosis through machine learning techniques. *Procedia Computer Science*, 112(C):2519–2528.

Mercaldo, F., Visaggio, C. A., Canfora, G., and Cimitile, A. (2016). Mobile malware detection in the real world. In *Software Engineering Companion (ICSE-C), IEEE/ACM International Conference on*, pages 744–746. IEEE.

Mitchell, T. M. (1999). Machine learning and data mining. *Communications of the ACM*, 42(11):30–36.

Miyajima, C., Nishiwaki, Y., Ozawa, K., Wakita, T., Itou, K., Takeda, K., and Itakura, F. (2007). Driver modeling based on driving behavior and its evaluation in driver identification. *Proceedings of the IEEE*, 95(2):427–437.

Nishiwaki, Y., Ozawa, K., Wakita, T., Miyajima, C., Itou, K., and Takeda, K. (2007). Driver identification based on spectral analysis of driving behavioral signals. In *Advances for In-Vehicle and Mobile Systems*, pages 25–34. Springer.

of Automotive Engineers, S. (1999). *On-Board Diagnostics for Light and Medium Duty Vehicles Standards Manual*. Society of Automotive Engineers, US.

Refaeilzadeh, P., Tang, L., and Liu, H. (2009). Cross-validation. In *Encyclopedia of database systems*, pages 532–538. Springer.

Taylor, A., Leblanc, S., and Japkowicz, N. (2016). Anomaly detection in automobile control network data with long short-term memory networks. In *Data Science and Advanced Analytics (DSAA), 2016 IEEE International Conference on*, pages 130–139. IEEE.

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

Family	Algorithm	FP Rate	Precision	Recall	F-Measure	Roc Area
All drivers	J48	0.001	0.992	0.992	0.992	0.998
	J48graft	0.001	0.992	0.992	0.992	0.998
	J48consolidated	0.001	0.991	0.991	0.991	0.998
	RandomTree	0.014	0.88	0.88	0.88	0.933
	RepTree	0.002	0.987	0.987	0.987	0.998
Driver A	J48	0	0.998	0.997	0.998	0.999
	J48graft	0	0.998	0.996	0.997	0.999
	J48consolidated	0.000	0.997	0.995	0.996	0.999
	RandomTree	0.004	0.956	0.944	0.95	0.97
	RepTree	0	0.997	0.996	0.996	1
Driver B	J48	0.001	0.991	0.994	0.992	0.998
	J48graft	0.002	0.99	0.994	0.992	0.998
	J48consolidated	0.002	0.990	0.990	0.990	0.998
	RandomTree	0.015	0.902	0.898	0.9	0.941
	RepTree	0.002	0.987	0.986	0.986	0.998
Driver C	J48	0.001	0.991	0.992	0.991	0.997
	J48graft	0.001	0.99	0.991	0.991	0.997
	J48consolidated	0.001	0.985	0.992	0.989	0.997
	RandomTree	0.015	0.823	0.826	0.824	0.905
	RepTree	0.002	0.977	0.979	0.978	0.998
Driver D	J48	0.002	0.991	0.988	0.989	0.996
	J48graft	0.001	0.992	0.987	0.99	0.996
	J48consolidated	0.002	0.988	0.981	0.984	0.997
	RandomTree	0.022	0.862	0.863	0.863	0.92
	RepTree	0.003	0.979	0.979	0.979	0.997
Driver E	J48	0	0.997	0.998	0.997	1
	J48graft	0	0.996	0.998	0.997	0.999
	J48consolidated	0.001	0.995	0.997	0.996	0.999
	RandomTree	0.005	0.949	0.954	0.952	0.974
	RepTree	0	0.996	0.997	0.996	0.999
Driver F	J48	0.001	0.994	0.996	0.995	0.999
	J48graft	0.001	0.993	0.996	0.994	0.999
	J48consolidated	0.001	0.992	0.994	0.993	0.999
	RandomTree	0.012	0.913	0.917	0.915	0.953
	RepTree	0.001	0.992	0.992	0.992	0.999
Driver G	J48	0.001	0.992	0.991	0.992	0.997
	J48graft	0.001	0.992	0.991	0.992	0.997
	J48consolidated	0.001	0.991	0.990	0.990	0.997
	RandomTree	0.01	0.881	0.885	0.883	0.937
	RepTree	0.001	0.984	0.984	0.984	0.998
Driver H	J48	0.001	0.993	0.993	0.993	0.998
	J48graft	0.001	0.993	0.993	0.993	0.998
	J48consolidated	0.001	0.990	0.990	0.990	0.998
	RandomTree	0.018	0.844	0.852	0.848	0.917
	RepTree	0.002	0.986	0.987	0.986	0.998
Driver I	J48	0.001	0.988	0.988	0.988	0.997
	J48graft	0.001	0.99	0.988	0.989	0.997
	J48consolidated	0.001	0.991	0.994	0.993	0.998
	RandomTree	0.018	0.808	0.816	0.812	0.899
	RepTree	0.001	0.988	0.984	0.986	0.998
Driver J	J48	0.001	0.99	0.988	0.989	0.997
	J48graft	0.001	0.99	0.99	0.99	0.997
	J48consolidated	0.001	0.990	0.990	0.990	0.998
	RandomTree	0.015	0.856	0.838	0.846	0.911
	RepTree	0.002	0.984	0.986	0.985	0.998

Table 1: Classification results: FP Rate, Precision, Recall, F-Measure and RocArea for classifying the full drivers dataset (multi-driver classification) and the single one computed with five different algorithms.