2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

Driver Profiling by Using LSTM Networks with Kalman Filtering

Adrian Kłusek[1], Marcin Kurdziel[1], Mateusz Paciorek[1], Piotr Wawryka[1] and Wojciech Turek[1]

***Abstract*— Nowadays, the most common way to model the driver behavior is to create, under some assumptions, a model of common patterns in driver maneuvers. These patterns are often modeled with averaged driver model. While this idea is very simple and intuitive in the context of driver classification by his/her patterns of maneuvers, our previous works demonstrated that assumptions underlying such models are often inaccurate and not applicable in general settings. In fact, it is very hard to express driving patterns with simple models. In this article we present a new way of modeling drivers: we employ Long-Short Term Memory networks to learn driver models from telematics data. In particular, our neural network models learn to predict driving-related signals, such as speed or acceleration, given the evolution of these signals up to the point of prediction. Solving this prediction task allows us to capture the behavioral model of the driver. We tested our models on several drivers, by predicting their future decisions. By learning our models on one driver and then evaluating them on another driver, we demonstrate that LSTM models are a powerful tool for driver profiling and detection of abnormal situations. We also evaluate the influence of data preprocessing on the quality of predictions. In this context we use Kalman filering, which can remove noise from uncertain dynamic measurements, in effect giving the best linearly estimated data.**

I. INTRODUCTION

During the last decade, smartphones became basic communication devices in everyday life. The most popular Android and iOS smartphones come with an array of integrated sensors, such as GPS receivers, accelerometers, gyroscopes, magnetometers, microphones and high-definition cameras. These sensors can be used as a low-cost physical measurement providers in a vast spectrum of different applications. Furthermore, smartphones can read and transmit data from a variety of Bluetooth-enabled external interfaces. In this work we focus on one particularly interesting avenue for applications of smartphone sensors, namely novel automotive active safety systems.

A. Motivation and related work

Active safety systems frequently rely on driver behavior prediction and anomaly detection. One common approach to this problem relies on building average driver profile. A drawback of this approach is that one model is shared by all drivers and used to detect anomalies. Anomaly detection in this context is a difficult problem, due to significant differences in driving styles between different drivers [3]. These driving style differences could be erroneously recognized as anomalous situations.

[1]AGH University of Science and Technology, al. A Mickiewicza 30, 30-059 Kraków, Poland klusek@agh.edu.pl

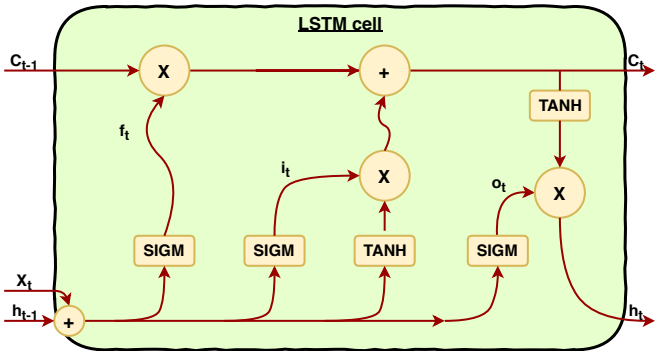

Fig. 1: *Structure of an LSTM cell.* ***X*** *input data,* ***h*** *hidden state,* ***c*** *memory cell,* f_t*,* i_t *and* o_t *represent the activations of forget, input and output gates, respectively.*

In this work we focus on building personalized driving models. Models of this kind can be used to detect driving patterns that do not correspond to a typical behaviour of the modelled driver. Such unusual driving patterns may indicate anomalous situations, such as, distracted driving. We attempt to *learn* personalized driving models from the data. In particular, our main contribution are fully unsupervised driving models implemented with modern recurrent neural networks. To construct such models we use an approach similar to *neural-probabilistic language models* [2], i.e. we cast the driver model estimation problem as a sequence learning task. Specifically, we use Long short-term memory (LSTM) networks [7] (Fig.1) to model sequences of important driving-related signals, such as speed, acceleration or engine load. During driving our trained models observe the driving-related signals and predict their future values in a short time window. A disagreement between the predictions and the actual measurements may indicate an anomalous driving situation. We present an experimental evaluation with real-world driving tests that confirm validity of our approach.

Our system uses a smartphone and a Bluetooth-connected OBDII interface to collect driving-related measurements. This in itself is not a novel idea. In fact, a number of recent works explore applications of smartphone-based sensing in active safety systems. The most important task in this context is the detection of dangerous driving [5], [6], [8], [9], [10]. Other interesting applications of smartphones in active safety systems include detection of distracted driving [12], [11], [13] and accident detection and notification [14]. Smartphones have also been used to construct driving models, which is a task most relevant to our work. An important

system of this kind with substantial real-world deployment, called *SafeDrive*, has been described in [15]. It is continuously recording basic driving characteristics read from an OBD interface and sends them via a mobile network to the central server. The collected data is fed to a driver behavioral model. This model is then used to detect anomalous driving. However, there are two main differences between *SafeDrive* and our approach. First, we attempt to construct personalized driving models whereas *SafeDrive* relies on a single model shared by all drivers. The second difference lies in the approach to driver modelling. *SafeDrive* employs a graph-based driver model, in which vertices represent driving stats and weighted edges represent possible transitions between these states. We, on the other hand, employ a modern recurrent neural network to learn the driver model in a fully unsupervised manner. This allows us to minimize prior structure enforced on the model. For example, we do not explicitly define driving states.

II. CONCEPT OF PROPOSED SYSTEM

A. Overall system structure

In order to perform the evaluation of the presented idea we have created a system consisting of two parts. The first part is responsible for collecting the data in the real traffic conditions. It uses a mobile phone with Android operating system and OBDII bluetooth interface. Detailed description of this part of the system and the collected data is presented in subsection II-B.

The second part of proposed system is an LSTM neural network launched on the remote server. Before the learning phase, we preprocessed the raw data to simplify the training of the LSTM models. After data preprocessing, the network was trained to predict the future values of the measured signals using values measured up to the point of prediction.

In order to evaluate the predictions, we introduced several numerical quality metrics presented in subsection II-E. We also employed charts to display the correspondence between the predicted data and the actual measurements.

B. Data description

The data used in the presented research has been collected during road experiments, which required creating a dedicated mobile laboratory. The laboratory was installed in a regular car in a way that allowed collecting the data during driving. The components of the laboratory are depicted in Fig.2. Details on the mobile laboratory building and testing can be found in [4] and in [3].

Three main sources of data were used:

1) car on-board computer, accessed via an OBDII interface,
2) mobile phone,
3) cameras.

The following data has been collected and stored every 200ms:

- timestamp,
- GPS position (longitude and latitude),

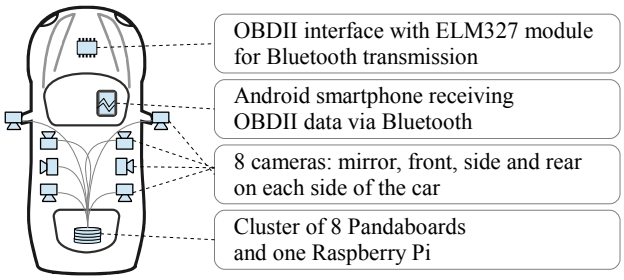

Fig. 2: Layout of the components installed in the car used for collecting the data [3].

- acceleration in Z axis (driving direction),
- brake pedal state,
- clutch pedal state,
- throttle state,
- engine revolutions per minute (RPM),
- covered distance and speed from odometer,
- torque,
- engine load,
- turn indicators state,
- images from 8 cameras.

The images from the 8 cameras were not used directly in the presented method. It helped to interpret several exceptional situations in the other types of data.

Fifteen volunteers took part in the conducted road experiments. They represented different age, sex and driving experience. Each driver was asked to repeat the same route several times – for each person at least five laps were recorded after at least two warm-up rides. The experiments were carried out in ordinary traffic, between 8:00 p.m. and 10:00 p.m. on business days.

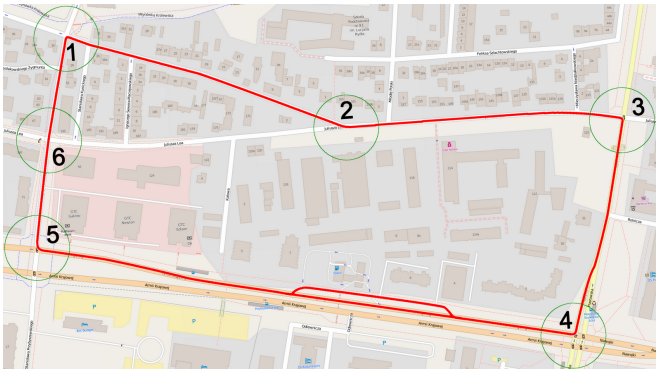

Fig. 3: The route used in the experiments with real drivers [3].

The route selected for the experiments (Fig. 3) was about 2 kilometers long and it contained 6 crossroads. Three of them were right-turns with traffic lights (numbers 3, 4 and 5 in Fig. 3). The other three situations were crossings without traffic lights.

C. Data preprocessing

Firstly, we chose subset of gathered data: acceleration in Z axis, engine revolutions per minute (RPM), speed (measured in kilometers per hour and captured from OBDII interface), torque and engine load. This set of parameters was chosen by noticing that the values of these parameters change very clearly in typical road maneuvers. Afterwards, the data for the LSTM-based sequence prediction needed to be scaled and preprocessed in order to achieve good prediction accuracy. We employed two-stage data preprocessing scheme. In the first step we normalized the data by using domain knowledge-based minimal and maximal values for each measured physical signal (Algorithm 1). These normalization ensures that input measurements for our model lay in the $\langle -1.0, 1.0 \rangle$, and therefore have comparable magnitudes. The normalization limits for the measured physical quantities are given in Table I. In the next step, we used Kalman filter to estimate noiseless measurements from the noisy measured signals.

Algorithm 1 Normalization algorithm

```
for i=1, K do
    for j=1, N do
      Data[i,j] = (data[i,j]-Min[i])/(Max[i]-Min[i])*2-1
    end for
end for
```

TABLE I: Maximal and minimal values for normalization of measured signals.

	Accel. (Z)	RPM	Speed	Torque	Engine load
Minimal	-1.0	0.0	0.0	-125.0	0.0
Maximal	1.5	4000.0	100.0	90.0	100.0

D. Driving style profiling

To model the individual driving styles, we use an LSTM network that takes as an input the measured driving signals and predict values of these signals in k future measurements. The parameters of built neural network are: batch size = 16, input/output size = 5, internal size = 128, learning epochs = 500, number of layers = 2, learning rate = 0.005 and sequence length = 8. This network is trained individually for each driver. The exact predicting algorithm is presented in Algorithm 2. In the first step, we set up the network by feed-forwarding a portion of the measured values. Next, we copy the state of the network for future use and predict k subsequent measurements starting from the current network state. After making the k predictions, we restore the network to the state before predictions and feed-forward next k actually measured values. This procedure is repeated for as long as new measurements are available.

E. Results evaluation

We evaluate usefulness of our method by comparing predictions of LSTM network on two types of test data. The first one is a set of measured values gathered for the same driver the network was trained on, but on different ride. The second type of data are measurements gathered for another drivers. For the first type of data predictions should be strongly correlated with measured data, whereas the second type should yield noticeable differences between measurements and predictions. Any discrepancy in first case should mean that the style of driving of tested driver has changed, which might suggest an abnormal situation. In order to quantify the correlations between the drivers, we propose one graphical and two numerical metrics. Note that we use these metrics with normalized measurements (see II-C).

Algorithm 2 LSTM prediction algorithm

```
# Set the network by first K
measurements:
for i=1, K do
    lastPred = network.forward(measurements[i])
    predictions[i] = measurements[i]
end for
# Split data into K-sized parts
for i = K+1 to measurements.size()-K, step = K do
    # Make the clone of the network
states
    networkClone = network.clone()
    # Forward on K predicted values
    for j = 0 to K-1, step = 1 do
        lastPred = network.forward(lastPred)
        predictions[i+j] = lastPred
    end for
    # Restore the network
    network = networkClone
    # Forward on k measured values
    for j = 0 to K-1, step = 1 do
        lastPred = network.forward(measurements[i+j])
    end for
end for
```

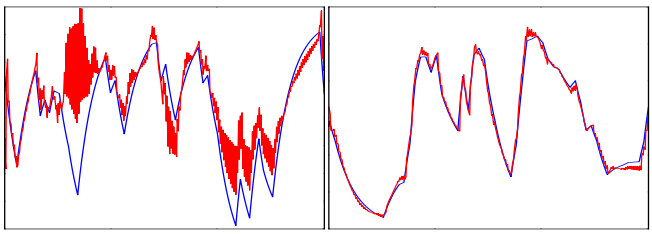

Fig. 4: Example model predictions for unrelated (left) and related (right) driving data. Blue line represents measurements, red line represents predictions.

The graphical comparison relies on analyzing the plots of measured and predicted values for each signal in the measured data. In situations where predictions are close to the measured values, the graphs present nearly overlapping lines (Fig. 4, right). When predictions are made for unrelated drivers (i.e. drivers other than the driver on which the model was trained), the prediction lines are noisy, struggling to fit

the measurements to the learned model (Fig. 4, left).

The first numerical metric is the average difference between the measured and the predicted values (SR), defined in Equation 1:

$$SR = \frac{1}{n}\sum_{i=1}^{n}(p_i - m_i)^2 * 100.0 \quad (1)$$

where p_i is $i-th$ predicted value and m_i is $i-th$ measured value.

The second numerical metric is the Pearson correlation coefficient [1] calculated as in Equation 2:

$$R_{pb} = \frac{\sum_i (p_i - \overline{p})(m_i - \overline{m})}{\sqrt{\sum_i (p_i - \overline{p})^2 \sum_i (m_i - \overline{m})^2}} * 100.0, \quad (2)$$

where p_i is $i-th$ predicted value, $\overline{p}$ is the mean of predicted values, m_i is $i-th$ measured value and $\overline{m}$ is the mean of measured values.

III. RESULTS

A. Modeling drivers behavior in specific road situations

The results presented in this subsection are trying to answer the question whether it is possible to use LSTM networks in modeling the trajectory of driving-related signals and distinguish drivers. We answer this inquiry positively by performing the experiment in real-world situation.

In the first step of this experiment, we have gathered the data on previously selected route presented in Figure 3 with several drivers and we chose one intersection (the 5th) on the selected route as a point where we are trying to predict the telematics data. As we mentioned in subsection II-C, in order to perform the test, we have limited our input to the LSTM networks only to the subset of gathered data types. In Figure 5 we present the achieved results for seven drivers. Each row corresponds to different driver's data and predictions and each column shows different type of data.

Please note that the network was trained on the trajectory which was collected by the last driver on different day to further diversify the data.

Taking under consideration both numerical and graphical metrics presented in subsection II-E we can ascertain that the LSTM networks are tools capable of modeling behavior of the drivers. We notice that looking only on the graphs we see that in each row (except the seventh one) there is more than one noisy plot where network struggled to fit predicted data (red lines) to the original data (blue lines). Judging only by that, we can suppose that the collected data from drivers A to F do not belong to the driver G. Only driver B look comparable to the driver G). We clearly see that the drivers A to F drove the vehicle in clearly different manner in the given intersection. The numerical results, shown in Table II, confirm that deduction. We start with the assumption that driver most fitting to the LSTM knowledge is the driver with the lowest average difference between the predicted and collected data and the highest value of Pearson correlation coefficient. Especially the acceleration and engine RPM clearly show that the driver G's behavior is the most similar to the data which was used to train the LSTM network. The speed and engine load values also help to distinguish between drivers, however with less clarity. From numerical point of view, the speed is the worst type of data to perform drivers distinction.

B. The influence of data preprocessing on model training and prediction results

In Figure 6 we present the influence of data preprocessing on model training and prediction in the processes of using LSTM neural networks. In each column we present each type of selected data: acceleration in Z axis, engine revolutions per minute (RPM), speed, torque and engine load. In each row we show data with no preprocessing and data with the Kalman filtering. We perform this test on the same trained neural network which was presented in previous subsection and we predict the data on the driver in the same crossroad (the 5th).

Comparing the graphs in Fig 6 and numerical measures in Table III we clearly see that application of Kalman filtering is the right choice for improving performance of the LSTM network. This is most clearly seen on first column which corresponds to the accelerometer. We can see that the noise is removed and the prediction error connected with the average falls down from 1.89 to 0.005. We also see the falling trend in the rest of the data. Moreover, utilization of the Kalman filter is noticeable in the Pearson correlation coefficient. In accelerometer, the value of this measure grows from 33.35 to 97.59.

IV. CONCLUSIONS

As the research has demonstrated we are able to successfully model individual driver's style with machine learning techniques. We manage to increase prediction accuracy with presented preprocessing employing Kalman filtering method. Proposed solution gives us possibility to recognize different driving styles. It also allows us to detect anomalies in driver's behavior in certain situations. These features might be applicable in many areas from automotive domain, e.g. active safety systems.

There is still plenty of room for improvement in the future. Better measurement metrics need to be found for more reliable verification of the quality of predictions. Different neural networks architectures should be considered to speed up computations and enhance results. Also, new preprocessing methods may have impact on learning process and the final outcome.

There is also a possibility to minimize the neural network and install it in personal device.

ACKNOWLEDGMENTS

The research presented in this article has received partial financing from the Polish National Center for Research and Development under the project no. TANGO2/340869/NCBR/2017. The research presented in this article has received partial financing from the AGH University of Science and Technology, Faculty of

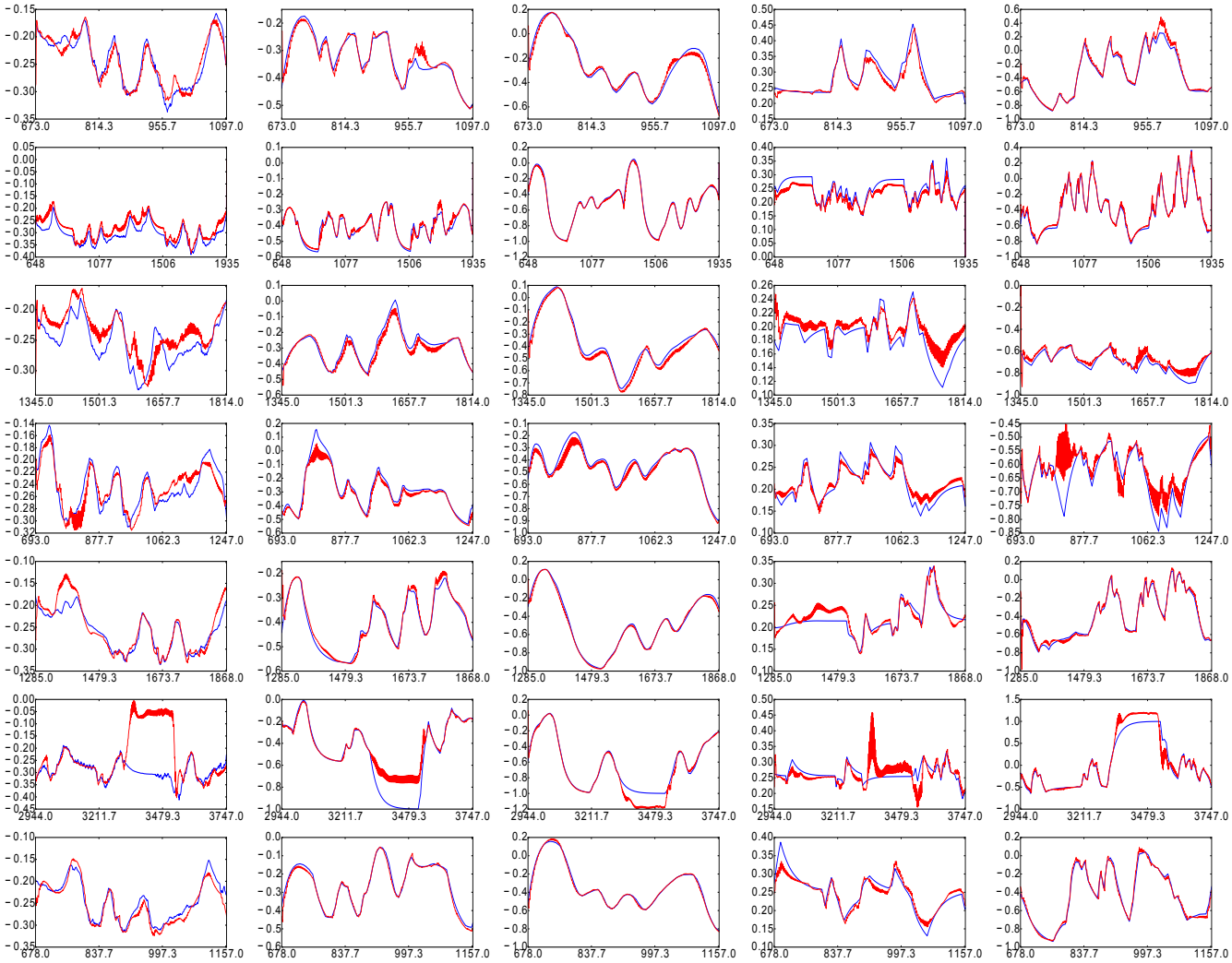

Fig. 5: Tests performed on 7 drivers: A, B, C, D, E, F, G. The net was trained on the driver G (another ride than presented). Each row represent one driver (A-G, from up to down). Columns from left to right represent: acceleration in Z axis, engine RPM, speed, torque and engine load.

TABLE II: Metric values for results shown in Fig 5.

Driver	Accel. (Z)	RPM	Speed	Torque	Engine load
A	SR: 0.024 R_{pb}: 94.02662	SR: 0.058 R_{pb}: 95.5923	SR: 0.138 R_{pb}: 98.49727	SR: 0.024 R_{pb}: 97.96944	SR: 0.284 R_{pb}: 99.48873
B	SR: 0.084 R_{pb}: 95.32351	SR: 0.036 R_{pb}: 97.52767	SR: 0.029 R_{pb}: 99.85385	SR: 0.048 R_{pb}: 95.7784	SR: 0.089 R_{pb}: 99.5767
C	SR: 0.072 R_{pb}: 80.67714	SR: 0.094 R_{pb}: 96.84817	SR: 0.153 R_{pb}: 99.03305	SR: 0.04 R_{pb}: 88.22306	SR: 0.385 R_{pb}: 81.48993
D	SR: 0.032 R_{pb}: 88.62257	SR: 0.168 R_{pb}: 98.59842	SR: 0.156 R_{pb}: 98.28886	SR: 0.026 R_{pb}: 94.90648	SR: 0.399 R_{pb}: 75.83605
E	SR: 0.048 R_{pb}: 94.01054	SR: 0.053 R_{pb}: 98.15856	SR: 0.069 R_{pb}: 99.69715	SR: 0.022 R_{pb}: 91.43173	SR: 0.218 R_{pb}: 98.34679
F	SR: 1.555 R_{pb}: 6.92539	SR: 1.586 R_{pb}: 97.34328	SR: 0.821 R_{pb}: 98.20366	SR: 0.112 R_{pb}: 35.77137	SR: 2.037 R_{pb}: 97.93792
G	SR: 0.023 R_{pb}: 95.55721	SR: 0.023 R_{pb}: 99.36985	SR: 0.033 R_{pb}: 99.74181	SR: 0.03 R_{pb}: 94.70787	SR: 0.08 R_{pb}: 99.51832

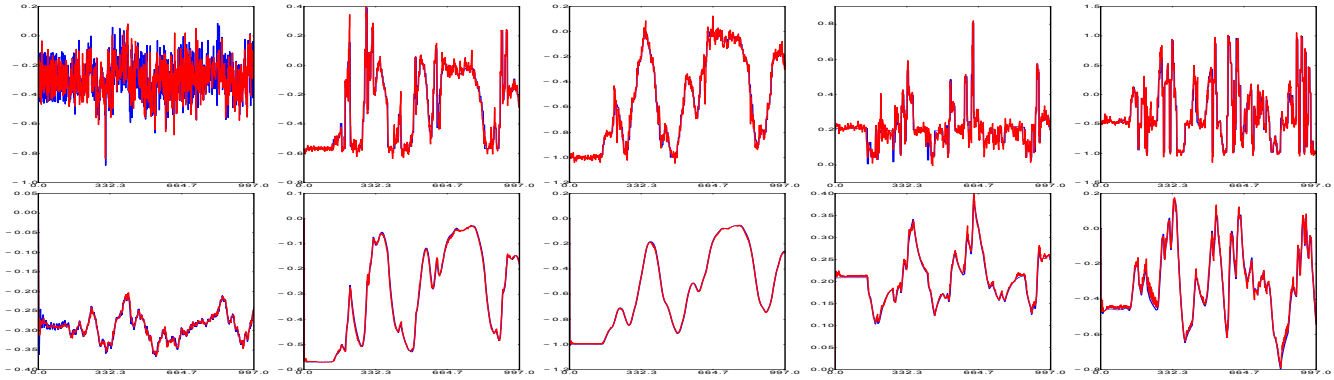

Fig. 6: Comparison of results for different data preprocessing. Each row represent one dataset with: no preprocessing and Kalman filtering. Each column represents: acceleration in Z axis, engine RPM, speed, torque and engine load.

TABLE III: Metric values for results shown in Fig 6.

Preprocessing method	Accel. (Z)	RPM	Speed	Torque	Engine load
None	SR: 1.892 R_{pb}: 33.35365	SR: 0.456 R_{pb}: 96.3781	SR: 0.236 R_{pb}: 99.07388	SR: 0.252 R_{pb}: 88.16916	SR: 5.281 R_{pb}: 88.76059
Kalman filtering	SR: 0.005 R_{pb}: 97.59523	SR: 0.005 R_{pb}: 99.9353	SR: 0.004 R_{pb}: 99.98733	SR: 0.003 R_{pb}: 99.64842	SR: 0.054 R_{pb}: 99.47506

Computer Science, Electronics and Telecommunications Dean's grant for Ph.D. Students. The research presented in this article has received partial financing from the AGH University of Science and Technology Statutory Project. The experiments have been conducted using PL-Grid Infrastructure (http://www.plgrid.pl/en).

REFERENCES

[1] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. Pearson correlation coefficient. In *Noise reduction in speech processing*, pages 1–4. Springer, 2009.

[2] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

[3] P. Blaszczyk, W. Turek, A. Byrski, and K. Cetnarowicz. Towards credible driver behavior modeling. In *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*, pages 1557–1562, Sept 2015.

[4] P. Blaszczyk, W. Turek, and K. Cetnarowicz. Extensible platform for studying the behavior of drivers in urban traffic. In *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, pages 1359–1362, Oct 2014.

[5] German Castignani, Thierry Derrmann, Raphaël Frank, and Thomas Engel. Driver behavior profiling using smartphones: A low-cost platform for driver monitoring. *IEEE Intelligent Transportation Systems Magazine*, 7(1):91–102, 2015.

[6] Haluk Eren, Semiha Makinist, Erhan Akin, and Alper Yilmaz. Estimating driving behavior by a smartphone. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 234–239. IEEE, 2012.

[7] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[8] Jin-Hyuk Hong, Ben Margines, and Anind K Dey. A smartphone-based sensing platform to model aggressive driving behaviors. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 4047–4056. ACM, 2014.

[9] Derick A Johnson and Mohan M Trivedi. Driving style recognition using a smartphone as a sensor platform. In *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pages 1609–1615. IEEE, 2011.

[10] Xinhua Liu, Huafeng Mei, Huachang Lu, Hailan Kuang, and Xiaolin Ma. A vehicle steering recognition system based on low-cost smartphone sensors. *Sensors*, 17(3):633, 2017.

[11] Xuefeng Liu, Jiannong Cao, Shaojie Tang, Zongjian He, and Jiaqi Wen. Drive now, text later: Nonintrusive texting-while-driving detection using smartphones. *IEEE Transactions on Mobile Computing*, 16(1):73–86, 2017.

[12] Homin Park, DaeHan Ahn, Taejoon Park, and Kang G Shin. Automatic identification of driver's smartphone exploiting common vehicle-riding actions. *IEEE Transactions on Mobile Computing*, 2017. accepted for publication.

[13] Yan Wang, Yingying Jennifer Chen, Jie Yang, Marco Gruteser, Richard P Martin, Hongbo Liu, Luyang Liu, and Cagdas Karatas. Determining driver phone use by exploiting smartphone integrated sensors. *IEEE Transactions on Mobile Computing*, 15(8):1965–1981, 2016.

[14] Jules White, Chris Thompson, Hamilton Turner, Brian Dougherty, and Douglas C Schmidt. Wreckwatch: Automatic traffic accident detection and notification with smartphones. *Mobile Networks and Applications*, 16(3):285, 2011.

[15] Mingming Zhang, Chao Chen, Tianyu Wo, Tao Xie, Md Zakirul Bhuiyan, and Xuelian Lin. Safedrive: Online driving anomaly detection from large-scale vehicle data. *IEEE Transactions on Industrial Informatics*, 13(4):2087–2096, 2017.