MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Driver Confusion Status Detection Using Recurrent Neural Networks

Hori, C.; Watanabe, S.; Hori, T.; Harsham, B.A.; Hershey, J.R.; Koji, Y.; Fujii, Y.; Furumoto, Y.

TR2016-088 July 2016

Abstract

In this paper, we present a method for estimating the confusion level of a driver using a classifier trained on multimodal sensor data. Using the driver confusion status detector, a car navigation system can proactively support the driver when he/she is confused. A corpus of data was collected during onroad driving in traffic using a navigation system and a car instrumented with a variety of sensors. The data was manually annotated with the driver's confusion status and with multiple features representing driver's behavior and the traffic conditions. We compared different types of classifiers trained from the data: logistic regression, a feed-forward neural network, a recurrent neural networks, and a long short-term memory (LSTM)-based recurrent neural network. The accuracy was evaluated using F-max as well as precision/recall. We found that the LSTM outperformed the other models.

IEEE International Conference on Multimedia and Expo (ICME)

DRIVER CONFUSION STATUS DETECTION USING RECURRENT NEURAL NETWORKS

Chiori Hori[1], *Shinji Watanabe*[1], *Takaaki Hori*[1], *Bret A. Harsham*[1], *John R. Hershey*[1],
Yusuke Koji[2], *Yoichi Fujii*[2], *Yuki Furumoto*[3],

[1]Mitsubishi Electric Research Laboratories,
Mitsubishi Electric Corporation
[2]Information Technology R&D Center
[3]Automotive Electronics Development Center

ABSTRACT

In this paper, we present a method for estimating the confusion level of a driver using a classifier trained on multimodal sensor data. Using the driver confusion status detector, a car navigation system can proactively support the driver when he/she is confused. A corpus of data was collected during on-road driving in traffic using a navigation system and a car instrumented with a variety of sensors. The data was manually annotated with the driver's confusion status and with multiple features representing driver's behavior and the traffic conditions. We compared different types of classifiers trained from the data: logistic regression, a feed-forward neural network, a recurrent neural networks, and a long short-term memory (LSTM)-based recurrent neural network. The accuracy was evaluated using F-max as well as precision/recall. We found that the LSTM outperformed the other models.

Index Terms— driver confusion status prediction, multimodal processing, recurrent neural network, long short-term memory

1. INTRODUCTION

Human-machine interfaces (HMI) for car information and entertainment systems are very important for safe driving and can offer a convenient interface to control navigation and other automotive functions. Speech interfaces are currently employed in car HMI's to reduce driving distraction. In practice, drivers need to handle complex situations inside and outside of the cars, such as difficult traffic conditions, unclear navigation instructions, and limited visibility. In such conditions, drivers may become confused because of a lack of information about how to proceed. Often, the needed information is available via the HMI, but the driver does not have enough time to retrieve that information using speech or manual interfaces. If the system can anticipate these situations then it can proactively provide more helpful information. We propose to detect driver confusion in order to provide a more proactive interface.

There has been some prior work directed at detecting the driver's state, or likely actions, using sensor data available in the vehicle. Available data may include traffic conditions, navigation status, vehicle status, and information about the driver's behavior that can be extracted from sensors such as cameras and microphones. In prior work, corpora of such data have been recorded during driving and annotated according to driver status and driving conditions [1, 2, 3]. In these studies, data-driven approaches were used for prediction. For example, the driver's emotional state was detected using a Bayesian network obtained from multimodal data consisting of traffic condition, driving condition, and the drivers' facial expressions [4]. Gaussian mixture models, estimated from speech signals [5], have been used for detection of driver stress. In addition, destination prediction and driver action prediction were investigated using driving condition histories, obtained using the controller area network (CAN) bus, and the navigation system status [6].

All of these approaches employed classification without modeling the dynamics of the signals. However, it has been suggested, in the context of stress detection in speech, that temporal dynamics of sensor data and the dependency between multiple features are important [7].

Recently, neural network models such as feed-forward deep neural networks (DNNs), recurrent neural networks (RNNs) and related architectures such as and long short-term memory (LSTM) RNNs, and convolutional neural networks (CNNs) have been shown to dramatically improve the performance of speech and image recognition. In addition, speaker emotion detection has been investigated in speech signals using RNN and LSTM models [8, 9].

The sensor data involved in driver state prediction is challenging due to the large variability and dynamic range. Deep network models may be more capable of modeling the dynamics and interdependencies in sensor data than previous approaches. In this study, therefore, we propose as a proof of concept to apply deep network architectures to the problem of predicting driver confusion. Since the complexity of the problem relative to the amount of data in our corpus is unknown, we compare performance using a variety of models: logistic regression (LR), DNNs, RNNs, and LSTM-RNNs.

2. MULTIMODAL DRIVER PREDICTION CORPUS

A corpus of data was collected during on-road driving in traffic using a car instrumented with a variety of sensors, including video cameras (focused both on the road and on the driver), microphones, car state sensors (via the CAN bus[1]), and a navigation system including global positioning system (GPS). From these a variety of time-series features were derived.

Eleven human subjects drove cars on a fixed route in a medium sized city in Japan. Drivers were guided by a car navigation system for the first section of the route and by a human navigator for the second section of the route. The two sections of the route were designed to require approximately equal driving time. The total driving time was approximately 55 minutes per driver. The data was manually annotated with the driver's confusion status as well as multiple features representing driver's behavior and the traffic conditions. Sensors and human annotations were sampled at one-second intervals. Table 1 shows the set of features used in our experiments along with their basic characteristics.

3. CLASSIFICATION MODELS

The four models we used for classification can all be seen as various forms of a neural network. Logistic regression is a simple log-linear model that corresponds to a simple feed-forward softmax "network" with no hidden layers. A more general architecture is shown in Figure 1. The network has an input layer that takes each input sensor data, a projection layer that reduces the multiple sensor information into a low-dimensional space, a hidden layer with recurrent connections that keeps context information, and an output layer that estimates a probability of driver confusion. The figure depicts a general RNN architecture, which incorporates left context via the recurrent connections (shown as a dashed directed edge). The DNN architecture can be seen as a special case of this model in which there are no recurrent connections. In the LSTM version of this model, the hidden layer units are LSTM cells instead of regular network units. The internal architecture of an LSTM cell is depicted in Figure 2. In theory, an LSTM cell can remember a value for an arbitrary length of time due to a system of gating. The LSTM cell contains input, forget, and output gates which determine when the input is significant enough to remember, when it should continue to remember or forget the value, and when it should contribute to the output value.

The input vector x_t is prepared as

$$x_t = \text{SensorInput}(t), \tag{1}$$

where SensorInput(t) is a set of multiple sensor data obtained at time t, and is converted to the input feature vector $x_t \in \mathbb{R}^N$.

[1]A controller area network (CAN bus) systems which is a vehicle bus standard designed to allow microcontrollers and devices to communicate with each other in applications.

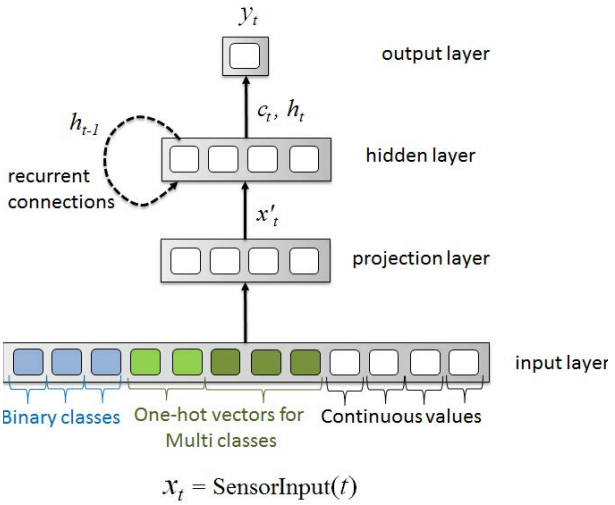

Fig. 1. Recurrent Neural Network.

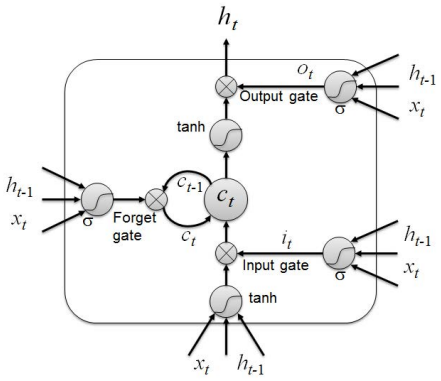

Fig. 2. LSTM cell.

The input vector is projected to the D dimensional vector

$$x'_t = W_{pr} x_t + b_{pr} \tag{2}$$

and fed to the recurrent hidden layer, where W_{pr} and b_{pr} are the projection matrix and the bias vector.

At the hidden layer, activation vector h_t is computed using the LSTM cells according to the way of [10][11], i.e.

$$\begin{aligned}
i_t &= \sigma(W_{xi} x'_t + W_{hi} h_{t-1} + b_i) && (3)\\
f_t &= \sigma(W_{xf} x'_t + W_{hf} h_{t-1} + b_f) && (4)\\
c_t &= f_t c_{t-1} + i_t \tanh(W_{xc} x'_t + W_{hc} h_{t-1} + b_c) && (5)\\
o_t &= \sigma(W_{xo} x'_t + W_{ho} h_{t-1} + b_o) && (6)\\
h_t &= o_t \tanh(c_t), && (7)
\end{aligned}$$

where $\sigma(.)$ is the element-wise sigmoid function, and i_t, f_t, o_t and c_t are the input gate, forget gate, output gate, and cell activation vectors for the sensor input at time t, respectively. In an abuse of notation, we identify the weight matrices and bias vectors via their indices. For example, W_{hi} is the hidden-input gate matrix and W_{xo} is the input-output gate matrix.

Table 1. Feature specification of multimodal sensed data. "$*$" shows the features already proposed in the existing systems. **H** and **S** in the last column show human annotation and sensor output, respectively. Colors correspond to those used in Figure 3.

Driving condition				
(1) Acceleration level*		Continuous values	$[0, 134]$	**S**
(2) Steering angle*		Continuous values	$[-6482, 6358]$	
(3) Velocity [km/h]*		Continuous values	$[0, 68.17]$	
(4) Gear shift position*		Multiple classes	6 levels	
(5) Wiper on/off*		Binary flag	0 or 1	
(6) Light on/off*		Binary flag	0 or 1	
(7) Turn signal on/off*		Binary flag	0 or 1	
(8) Driving action		Multiple classes	9 types : change lanes, stop, U-turn, backward, turn left*, right/left curve*, passing parking cars on the street, others	**H**
Traffic condition				
(9) Road type		Multiple classes	12 types : highway, urban express way, national road, main local road, prefectural road, regular road type I and II, approach ramp, secondary street type I and II, ferry route, others	**H**
Car ahead	(10) Type	Multiple classes	8 types : bus, truck, motor bike, bicycle, light car, standard-sized car, others	
	(11) Color	Multiple classes	10 types : white, red*, black, blue, yellow, green, silver, gray, brown, others	
(12) Traffic signals		Multiple classes	5 types : blue, red, yellow, directed one way, others	
(13) Number of lanes		Multiple classes	4 types : one lane, two lanes, w/o center line, others	
(14) Lane type		Multiple classes	4 types : two-lane road without a dividing strip, two-lane and four-lane road for two ways, others	
(15) Oncoming car*		Binary flag	0 or 1	
(16) Cars parking on the street*		Binary flag	0 or 1	
Driver's behavior				
Voice activity	(17) Driver	Binary flag	0 or 1	**H**
	(18) Human Navigator	Binary flag	0 or 1	
	(19) Navigation system	Binary flag	0 or 1	
Gazing direction	(20) Simple	Multiple classes	8 types : staring forward, backward with tour around, right or left backward, room mirror, right or left mirror	**H**
	(21) Complex	Multiple classes	2 types : looking around, bent forward	
(22) Other Behaviors		Multiple classes	2 types : raising hands, bent forward	
Location information				
(23) Checkpoints on the route		Multiple classes	6 types	**S**
(24) Longitude of the car		Multiple classes	Continuous values	
(25) Latitude of the car		Multiple classes	Continuous values	
(26) Distance to the goal		Multiple classes	Continuous values	

The output value at time t is computed as

$$y_t = \sigma(W_{HO} h_t + b_O), \tag{8}$$

where W_{HO} and b_O are the transformation matrix and the bias vector to classify the input vector into binary classes, i.e. confused or not confused, according to the hidden vector. The sigmoid function is used to normalize output values so that they range from 0 to 1.

The final output of the network can be considered as a confusion probability of the driver at time t. If the probability is greater than a threshold, the system detects confusion status. The sensor data are sequentially input to the network and a sequence of confusion probabilities is obtained, one for each time t.

4. EXPERIMENTS

4.1. Data

We evaluated driver's confusion detection using multiple sensor data from 11 drivers. Because of the small amount of data, we used a leave-one-out evaluation approach. We used each driver's data as a test set, for which we built a driver-independent classifier trained on data from eight of the other drivers, for each type of model. The data from the two remaining drivers were used for development, for example, parameter tuning and early stopping. The resulting classifier was evaluated on the held-out driver's data.

There were approximately 50 times as many negatively labeled (non-confusion) samples as positively labeled (confusion) samples.

We selected a subset of the sensors and driving conditions in a preliminary experiment by individually testing the performance of each component as a predictor of driver confusion. The set of features used for the experiments is listed in Table 1. Fig. 3 shows the maximum F-score (MaxF) of a set of Logistic Regression classifiers for the confusion probability, individually trained on each component, sorted by maximum F-score. The maximum F-score for each classifier was calculated as the maximum of the F-scores from the precision and recall rates obtained across all detection thresholds for that classifier. The maximum F-scores give some indication of how predictive the feature is for our data set.

In Fig. 3, location information shows the highest scores, followed by traffic condition, driving status, driver's condi-

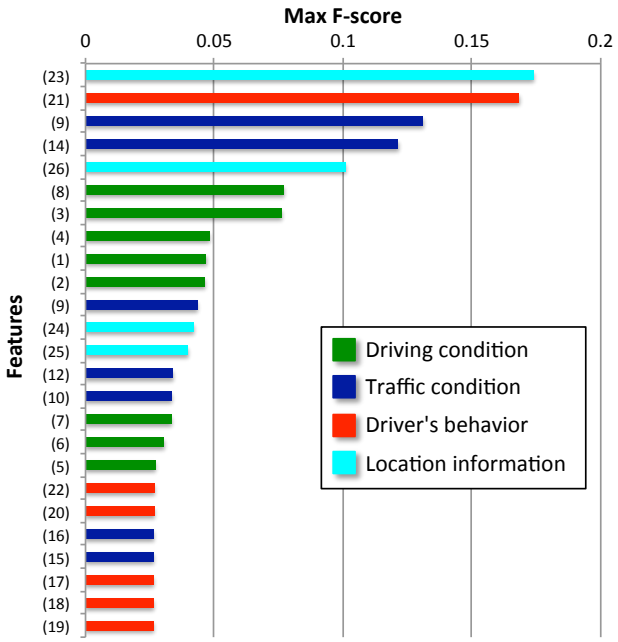

Fig. 3. Effective sensors and driving conditions for the features listed in Table 1, indexed by (1) through (26). The horizontal axis is the maximum F-score (MaxF) where the maximum is taken over all threshold settings.

tion, and finally navigation status. The importance of location features indicates that there were some points in the route where many drivers were confused. Although the location appears to be informative for prediction of confusion, we excluded it from the set of input features for subsequent experiments because we are interested in features sensitive to the characteristics of those locations rather than the locations themselves. From the point of view of generalization, we would need data from every possible location to generalize on the basis of location features. Further study is needed to understand the characteristics that make a location confusing. With the exception of location, we selected the top scoring 38 feature components for training the classifiers. These components are listed in Table 1.

Table 2. Evaluation results of the models using Max F-score (MaxF) and Average Precision (AP).

	MaxF		AP	
	Dev. set	Test set	Dev. set	Test set
LR	0.22	0.23	0.11	0.12
DNN	0.25	0.26	0.14	0.17
RNN	0.28	0.26	0.16	0.17
LSTM	**0.30**	**0.31**	**0.19**	**0.21**

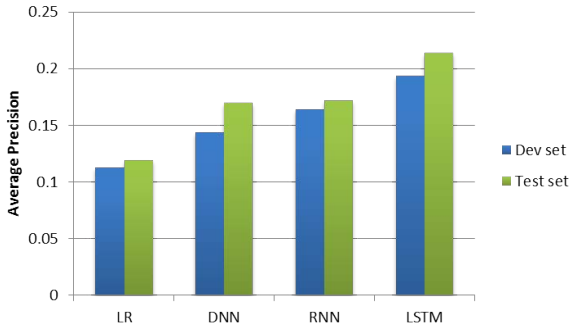

Fig. 4. Comparison of Average Precision among the methods.

4.2. Classifiers

To evaluate the efficiency of our proposed method, we compared confusion detection performances of four models, LR, DNN, RNN, and LSTM RNN. L2 regularization using weight decay was used for the LR model. The other models were not regularized. The LR and DNN systems used sensor data independently at each time frame. The DNN had two hidden layers, each of which had 100 units. The simple RNN and the LSTM RNN had one projection layer and two hidden/LSTM layers, each of which had 100 units. The output layer of each network had only one unit with a sigmoid activation, which indicates a probability of driver confusion. All the neural networks were trained using the stochastic gradient decent method, where the development set was used to select the best training parameters.

For both training and testing the RNN and LSTM RNNs, sequences of 20 frame input chunks were used with a 10 frame left context "warmup" period. In pilot experiments this yielded similar results to testing on one long input sequence. The LR, DNNs, RNNs and LSTMs were trained using the Chainer neural network toolkit [12].

4.3. Evaluation Metrics

The max F-score (MaxF) and the average precision (AP) were used for the evaluation. MaxF corresponds to maximum value of F-score on the precision/recall curve. The average precision corresponds to the area under the precision/recall curve. As noted above, our data has a large imbalance between positive and negative samples, and in general we would expect this to be true for driving - drivers are in a normal (negative confusion) state much more often than they are in a confused state. It is well known that ROC curves can be misleading for imbalanced data, because the useful operating region for the model is a very narrow area at one edge of the ROC curve. Therefore, we use PR curves and Average Precision instead for evaluation.

4.4. Evaluation Results

The experimental results are shown in Table 2 and Fig. 4. The models were selected based on the best AP for the dev

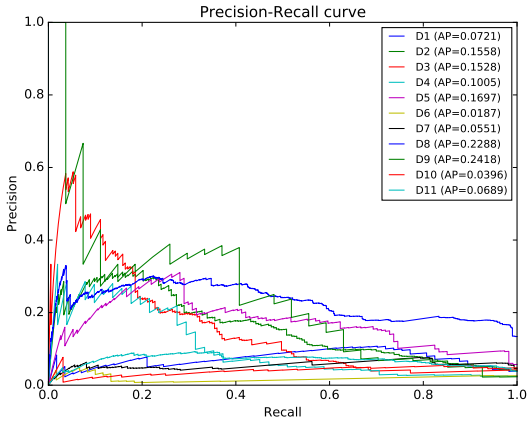

Fig. 5. Precision/Recall curve for LR.

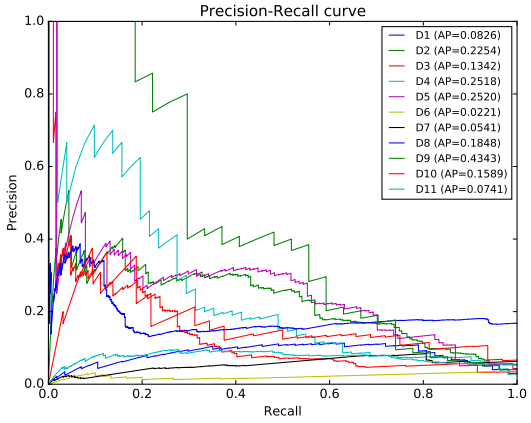

Fig. 6. Precision/Recall curve for DNN.

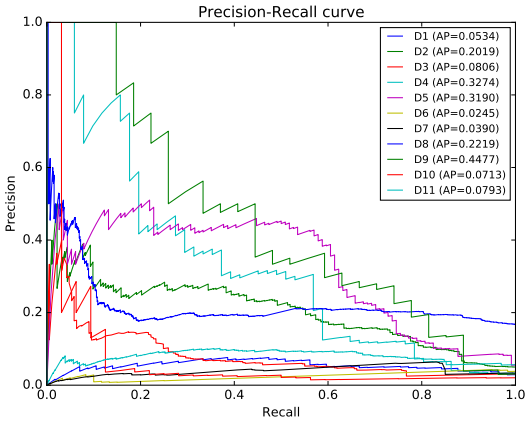

Fig. 7. Precision/Recall curve for RNN.

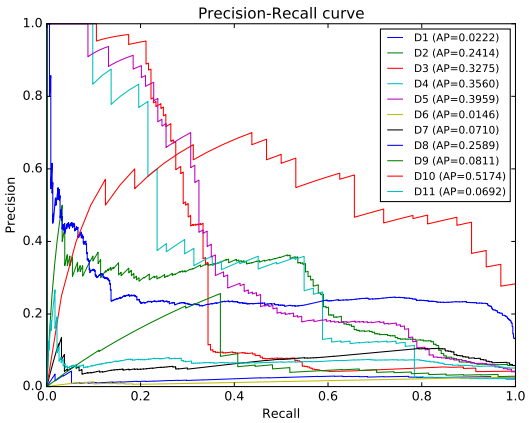

Fig. 8. Precision/Recall curve for LSTM.

set. The values in the table represent the max F-score (MaxF) and the average precision (AP) for each classifier. For the LR model, we obtained 0.23 max F-score for the test set. Next we tested DNNs and obtained 0.26 max F-score, which is larger than LR. The simple RNN classifier performance was comparable to that of the DNN. Finally, we tested the LSTM RNN and obtained a significant improvement to 0.31. We also measured AP. The average precision corresponds to the area under the precision/recall curve. Similarly to the MaxF results, the LSTM RNN significantly outperformed the other classifiers. This improvement is probably due to the LSTM's greater ability to make use of long time context.

However, the MaxF and AP values obtained in these experiments are still not high enough to use for actual applications, but show some promising predictability given the small data size. Driver-dependent factors may contribute too much variance relative to the detectable confusion signal, but such problems may be greatly diminished with larger training data.

This can be seen in the performance for individual drivers. Figures 5-8 show the precision/recall curves for all drivers, one figure per classifier. The performance of the different models for different drivers can be compared in Figure 9, which shows the AP score for each driver for each model. Here we compare an additional type of LSTM, trained and tested in 10-frame chunks (LSTM-10) in addition to the standard LSTM with 20-frame context used elsewhere. The performance of the models varied greatly between drivers. The confusion status was moderately detected for some drivers, while it was detected poorly for Drivers 1, 6, 7 and 11. Driver 9 was the best case for the LR, DNN and RNN classifiers, but the 20-frame LSTM classifier performed much worse than the 10-frame version for Driver 9. For Driver 5, the reverse was true: the performance of the LSTM model was good, while LSTM-10 had much worse performance. On the other hand, for Driver 10, both LSTM classifiers had good performance while the other classifiers performed poorly. Apparently variation in the temporal context dependencies of driver behavior may cause the optimal LSTM context size to differ between drivers. If this is true, then a mismatch between training and test drivers could give rise to the difference in performance

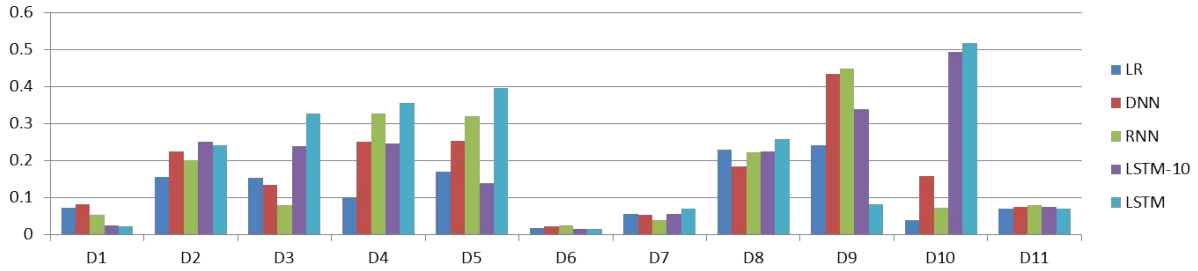

Fig. 9. Comparison of Average Precision among the drivers.

seen for Drivers 5 and 9 LSTM's. Perhaps this could be mitigated by allowing the model to handle context in a more flexible way between drivers, or it might just reflect an idiosyncrasy of a relatively small data-set.

Overall, the wide range of performance across drivers indicates that some driver dependency needs to be incorporated into the models to improve the performance for many drivers. We believe that using a corpus with both more drivers, and much more data for each driver would help the models incorporate various drivers' behaviors. Additionally, model parameters could be adapted to incorporate different context sensitivities for each driver.

5. CONCLUSION

We applied several different classifier types to driver confusion detection using data collected from heterogeneous sensors and driving conditions in real driving. The LSTM RNN outperformed logistic regression, feed-forward neural networks and simple RNNs. This may be because of the LSTM's greater ability to make use of long time context, and may indicate that such context is important to detect driver's status. Future work will include LSTMs trained using features from more sensors including audio, video and other recognition systems' outputs.

6. REFERENCES

[1] Nobuo Kawaguchi, Shigeki Matsubara, Hiroyuki Iwa, Shoji Kajita, Kazuya Takeda, Fumitada Itakura, and Yasuyoshi Inagaki, "Construction of speech corpus in moving car environment," in *Sixth International Conference on Spoken Language Processing in Beijing China on October 16-20 2000 (ICSLP 2000). v. 3, 2000, p. 362-365*, 2000.

[2] Teruhisa Misu, Antoine Raux, Ian Lane, Joan Devassy, and Rakesh Gupta, "Situated multi-modal dialog system in vehicles," in *Proceedings of the 6th workshop on Eye gaze in intelligent human machine interaction: gaze in multimodal interaction*. ACM, 2013, pp. 25–28.

[3] David Cohen, Akshay Chandrashekaran, Ian Lane, and Antoine Raux, "The HRI-CMU corpus of situated in-car interactions," *Proc. IWSDS*, pp. 201–212, 2014.

[4] Lucas Malta, Chiyomi Miyajima, Norihide Kitaoka, and Kazuya Takeda, "Analysis of real-world driver's frustration," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, no. 1, pp. 109–118, 2011.

[5] Texus Univ., "Driver," *Intelligent Transportation Systems, IEEE Transactions on*, 2009.

[6] Bret Harsham, Shinji Watanabe, Alan Esenther, John Hershey, Jonathan Le Roux, Yi Luan, Daniel Nikovski, and Vamsi Potluru, "Driver prediction to improve interaction with in-vehicle HMI," in *Proc. Workshop on Digital Signal Processing for In-Vehicle Systems (DSP)*, Oct. 2015.

[7] Raul Fernandez and Rosalind W Picard, "Modeling drivers speech under stress," *Speech Communication*, vol. 40, no. 1, pp. 145–159, 2003.

[8] Martin Wöllmer, Florian Eyben, Stephan Reiter, Björn Schuller, Cate Cox, Ellen Douglas-Cowie, and Roddy Cowie, "Abandoning emotion classes-towards continuous emotion recognition with modelling of long-range dependencies.," *INTERSPEECH*, vol. 2008, pp. 597–600, 2008.

[9] Martin Wöllmer, Björn Schuller, Florian Eyben, and Gerhard Rigoll, "Combining long short-term memory and dynamic bayesian networks for incremental emotion-sensitive artificial listening," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 4, no. 5, pp. 867–881, 2010.

[10] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, May 2013, pp. 6645–6649.

[12] Preferred Networks, "Chainer," in *"http://chainer.org/"*.