

# Brake Maneuver Prediction – An Inference Leveraging RNN Focus on Sensor Confidence

Shu Liu<sup>1</sup>, Kevin Koch<sup>2</sup>, Bernhard Gahr<sup>2</sup> and Felix Wortmann<sup>2</sup>

**Abstract**—In recent years, driver behavior analysis has led to countless driver assistance systems. In these systems, earlier detection of a driver’s maneuver intentions offers opportunities to improve driving experience and safety. Especially brake maneuvers are of fundamental importance because they are directly related to the avoidance of potential hazards.

Current state-of-the-art brake assistance systems rely on the release speed of accelerator pedal as an indicator whether a brake event is planned. However, this simple and practical algorithm, fails to capture the overall movement pattern of accelerator pedal behaviors and cannot utilize rich information from different vehicle sensors.

To address this issue, we propose a novel recurrent neural network architecture for the purpose of brake maneuver prediction. The proposed method exploits the advantages of multiple sensors. Unlike conventional practices where all signals are aggregated to a single neural network, we leverage the confidence of each sensor. We evaluate our approach based on a dataset of 44 drivers, comprising around 500 hours of naturalistic driving data. The evaluation results show that the proposed algorithm outperforms baseline method by large margin.

## I. INTRODUCTION

The transportation safety is of significant importance. Each year, around 20 to 50 million people suffer from traffic accident, with more than 1.2 million people killed in traffic collisions [1]. There are two basic strategies to avoid collisions, namely steering to evade or braking. To perform evasion without causing any additional risk to traffic flow, highly developed driving abilities and high situation awareness are required [2]. Many researches also showed that in critical situations, braking is favored than steering [3], [4]. Moreover, despite of advanced sensing and control techniques, Advance Driver Assistance System (ADAS) with emergency steering for collision avoidance is not yet available for public [5]. Additionally, drivers tend to perform actions that are against automatic steering assistance at subconscious level [6]. Overall, the applicability of steering interventions have a limited agreement with drivers whereas braking is in favor.

There are essentially two types of methods to improve the driving safety via brake. The first type is emergency brake systems, which rely on external cameras, radar, or lidar to detect ambient obstacles and imminent collisions and apply automatic brake maneuver. The other type is brake assistance

systems that infer hard brake intention from driver behaviors and prepare deceleration in advance or reinforce the brake if the driver fails to press the brake pedal hardly enough. In this paper, we focus on the improvement of the later one, the brake assistance system. In brake assistance systems, one of the inferences of the driver’s intention is based on release speed of the accelerator pedal, since in the event of potential collisions drivers usually release the accelerator pedal faster than normal and slam the brake pedal [7]–[10]. Existing solutions apply negative torque measures prior to brakes, for example, by controlling the engine air intake, the electric motor, or by closing the gap between brake disc and pads [9]. These measures can shorten the brake preparation time and thus the braking distance.

Brake assistance systems are advantageous in several perspectives. In addition to the improved safety, such systems are tolerant to false alarm. It is worth noting that applying negative torque measures to the wheels does not result in an automatic brake. In the case of a false alarm, brake assistance systems only create minor deceleration. This property favors the acceptance of consumers and avoids that alarm mistrust leads to drivers’ rejection of assistance [11], [12].

Nevertheless, an improved inference accuracy is beneficial. First, it increases the sustainability and the efficiency of vehicles. An imprecise system causes undue wear to brake systems. For example, friction elements may have diminished braking capacity; false application of negative torque causes unnecessary fuel consumption. Furthermore, anticipating brake maneuver can also benefit other traffic participants, for example, by turning on brake lights or broadcasting maneuver information via V2V communication [13]–[15].

We interviewed long-term brake assistance system researchers from an industry-leading automotive supplier and confirmed that conventional accelerator actuated emergency brake systems are triggered when release speed of accelerator exceeds a predefined threshold. However, this approach entails some inherent shortcomings. First, the motion pattern of accelerator pedal contains more information than just its speed. Second, in modern vehicles, heterogeneous sensor information can be queried, which provides a comprehensive picture of the state of vehicle and driver. Nevertheless, it is challenging to manually build a rule-based algorithm that can model complex information.

Neural networks are a popular technique to build models with high expressiveness. There are many researches that use neural networks to process CAN-Bus data, e.g. [16], [17]. Most of these works aggregate all signals to one neural net-

\*This work was supported by Robert Bosch GmbH

<sup>1</sup>Department of Management, Technology and Economics, ETH Zurich, Switzerland liush@ethz.ch

<sup>2</sup>Institute of Technology Management, University St. Gallen, Switzerland {kevin.koch, bernhard.gahr, felix.wortmann}@unisg.ch

work, aiming to model complex relations between sensors. However, such design has the risk that neural networks overfit to the relations that do not generalize as we will show in the following.

To predict brake maneuvers, we have found that each sensor has a different level of confidence in certain scenarios. Using this feature an improved performance can be achieved compared to modeling the relations between different sensors. The paper is structured as follows: in Sec-II, related researches on driver maneuver prediction are described. The field test settings and model design are presented in Sec-III. In Sec-IV, we present the detailed analysis of the proposed algorithm and the comparison with the state-of-the-art method. In the final section, we discuss potential improvements.

Overall, the main contributions of our paper are:

- We first demonstrate that even basic recurrent neural network (RNN) can outperform existing industrial solutions.
- We propose a novel recurrent neural architecture that leverages confidence of different vehicle sensors and further improves the performance.
- We provide comprehensive validation of the proposed method using a naturalistic dataset.
- Our proposed method is based exclusively on CAN-Bus data without any additional sensory and can be readily deployed in current vehicles.

## II. RELATED WORK

A large body of research in driver behavior analysis and maneuver prediction relies on inside monitoring cameras. The cameras capture the drivers behaviors such as head pose, gaze, or foot movement to infer maneuvers as braking, acceleration, or turns [18]–[22]. These methods, despite their proven effectiveness, are yet seldom adapted because monitoring cameras are uncommon in vehicles. Moreover, additional supervision by camera can raise privacy issue.

In practice, a common method for the inference of brake intention is to monitor the accelerator pedal as several patents show. In [7], [8], the inventors presented a process that executes automatic braking operation, when the return speed of the accelerator pedal exceeds a threshold. In [9], the patent holder uses a sensor configured to determine the rate of travel of the accelerator pedal. In the case of the sensor detecting a quick release of the accelerator pedal, the control module of a vehicle checks other vehicle sensors that could indicate a braking event. If other vehicle sensors also report an alarm, negative torque is applied to wheels. In [10], the braking force is a function of the speed applied by the driver to the accelerator pedal and the relative position of the accelerator pedal. These widespread methods only included a few information about the accelerator pedal but not the complete movement pattern. In addition, [20]–[22] showed that monitoring foot movement of drivers contributes to an overall higher accuracy of brake maneuver prediction.

The latest work from Hallac et al. [17] takes a different approach than focusing solely on braking detection. From a

large number of CAN signals, the authors segment individual events by compactly displaying the signals using a RNN as encoder. Hard brake events are one of the categories. Their approach also shows that the large number of CAN signals makes it possible to reliably predict future CAN signal values. However, their approach lacks a specific application scenario and is mostly exploitative. Our research should combine both approaches. We aim at a specific use case and apply the strength of RNN with sensor confidence to perform brake maneuver forecasts based on existing sensors. There are researches [23]–[26] that demonstrate modeling confidence and uncertainties can improve neural network performance; besides, in both theory and practices, e.g. [27]–[29], ensemble of multiple estimators (in our context, different sensors) can generally improve prediction performance. Our proposed method differs in that we focus the on confidence of sensor inputs and hence provide a novel ensemble method for sensor fusion. Furthermore, we explore and provide explainable reasons why the proposed method outperforms rule-based algorithm and neural network approaches that do not consider sensor confidence.

## III. DATASET DESCRIPTION

**Data Acquisition.** Our research is based on a dataset from a large naturalistic field study in cooperation with a national road assistance service. In this field study, 50 professional road assistance drivers drove over a period of three months on real roads in Switzerland. Total driving distance accumulated to approximately 300,000 kilometers. There was no limitation on drivers about the routes they chose or any special tasks they should perform. During the drives, CAN-Bus data was accessed and collected via the OBD-II interface. All vehicles were of the same type. The sensors have different sample frequency and as a signal preprocessing step we resample and interpolate them to 10 Hz. After data cleansing, 44 drivers with on average around 12 hours of driving data was used.

Among all sensors, we choose the following signals for brake maneuver prediction: speed values of all 4 wheels, accelerator pedal position, brake pedal position, longitudinal and lateral acceleration. Brake pedal position is only used to annotate hard brake events but not used as algorithm input. The reason for using only a small subset of all CAN-Bus signals are: a) the chosen sensors are the most common ones across different types of vehicles and this ensures the generalization ability of the algorithm; b) the chosen sensors are most representative for the dynamics of a vehicle before hard brakes; by restricting to these sensors, we make sure our neural network is not over-fitting to vehicle specific sensors that are coupled to accelerator or brake pedal; c) a small subset of signals allows us to increase computation efficiency for time critical tasks.

**Event Segmentation.** We first introduce the definitions of *brake event* and *accelerator event*. Sensors like brake pedal and accelerator pedal can be differentiated into *on* and *off* states. *Off* states present the period when the pedal is released completely and *on* states present the period when

acceleration or brake maneuvers are operated. The *on* states can be segmented using the algorithm proposed in [30].

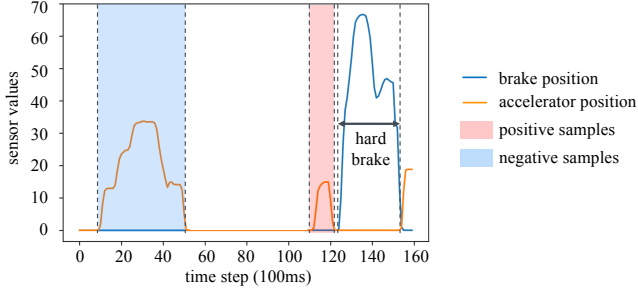


Fig. 1: Definition of positive and negative samples

**Positive and Negative Samples.** We first define brake slam as the moment when the brake pedal position increases over more than 25% of its total range within 0.1s. We only consider the brakes as hard brakes if a slam happens in the initial part (i.e. before the first derivative of brake position reaches zero) of a brake event. There are brake events in which slams happen in the later part of the event. However, a brake slam in the later part is not of important significance, because in such case, a vehicle is already prepared for brakes. Based on the above definitions, we derive positive and negative samples as follows (also shown in Fig. 1):

**Positive samples:** accelerator events that are followed by hard brakes and the brake slam happens within certain *time gap* after the release of accelerator pedal.

**Negative samples:** accelerator events that are not followed by any brake events within *time gap*.

We choose the length of the *time gap* as 1.0s. It is noteworthy that accelerator events followed by soft brakes are not of our interest, because in such case, a true positive alarm does not contribute significantly to safer driving. Compared to [17], there are significantly more positive examples in our data set (368 positive examples in around 520 hours vs. 122 positive examples in 2098 hours of driving data). Our hard brake definition is slightly more relaxed to focus not only on extreme events. To evaluate generalization capability of algorithms, we perform all experiments 10 times by randomly splitting data into train (36 drivers) and test (8 drivers) sets.

## IV. METHODS

### A. Baseline Method

According to the description in [7]–[10] and our interview with brake system experts from a industry-leading automotive supplier, we rebuild the Algorithm 1 for the detection of rapid accelerator pedal release. The algorithm iterates over all signals of an accelerator event  $\vec{X}$ . If within a selected time *interval*, the release speed of the accelerator pedal is above a certain *threshold*, the algorithm returns true and brake preparation measures would be executed. If the threshold is never exceeded, the algorithm returns false and there is no brake preparation. *threshold* and *interval* are tuned via grid

search and the optimal values are *threshold* = 6% of total brake pedal range with *interval* = 0.1s.

---

### Algorithm 1 Accelerator Pedal Rapid Release Detection

---

```

1: Input:  $\vec{X}$       ▷ Input is a segmented accelerator event
2:  $N = \text{len}(\vec{X})$ 
3:  $flag\_threshold = \text{False}$ 
4: for  $i = interval : N$  do
5:    $release\_speed = \vec{X}[i - interval] - \vec{X}[i]$ 
6:   if  $release\_speed > threshold$  then
7:      $flag\_threshold = \text{True}$ 
8: if  $flag\_threshold$  then
9:   return True
10: else
11:  return False

```

---

### B. Proposed Method

**Input and Feature Engineering** The average length of an acceleration event is around 12s in our dataset. Since the most relevant information about an upcoming brake lies in the later part of an acceleration event, we only take the last 3s of an event as input. We have found that a longer input sequence does not contribute to better performance. We choose the following sensors and their first and second derivatives as input:

- Accelerator pedal position (1×3)
- Longitudinal and lateral acceleration (2×3)
- Steering wheel angle (1×3)
- Speeds of four wheels (4×3)

**Recurrent Architecture** Recurrent neural networks (RNN) are widely use for modelling of contextual temporal information. Unlike feed-forward neural architectures that compute an output  $\vec{o}(t)$  at each time step, RNNs compute an additional hidden state  $\vec{h}(t)$  that function as a task-relevant summary of the past input sequence  $\vec{x}(0) \dots \vec{x}(t)$  up to time  $t$ . Various of variants of RNN such as Gated recurrent unit (GRU) [31] and Long short-term memory (LSTM) [32] are proposed that can better handle information across large time scale. In this paper we use a specific form of RNNs, namely GRU. The advantage of GRU over RNN is that a GRU has an update gate  $\vec{z}(t)$  and a reset gate  $\vec{r}(t)$  that control how much influence the current input  $\vec{x}(t)$  and the previous hidden state  $\vec{h}(t-1)$  has on the next hidden state  $\vec{h}(t)$ . The update procedure is as follows:

$$\vec{z}(t) = \sigma(\mathbf{W}_z \vec{x}(t) + \mathbf{U}_z \vec{h}(t-1)),$$

$$\vec{r}(t) = \sigma(\mathbf{W}_r \vec{x}(t) + \mathbf{U}_r \vec{h}(t-1)),$$

$$\vec{h}(t) = \vec{z}(t) \circ \vec{h}(t-1) + (1 - \vec{z}(t)) \circ \sigma(\mathbf{W}_h \vec{x}(t) + \vec{r}(t) \circ \mathbf{U}_h \vec{h}(t-1)),$$

where  $\mathbf{W}_{z,r,h}$  and  $\mathbf{U}_{z,r,h}$  are GRU parameters and  $\sigma$ ,  $\circ$  denote *tanh* function and element-wise product respectively.

One of the basic approach to model temporal information is using GRU as an encoder. In practices, to utilize the dependence of different sensors, all signals are concatenated

and fed into one GRU, we denote this approach with superscription (*all*). A sequence of signal  $\vec{x}^{(all)}(1, \dots, t)$  is fed into a GRU. At the end of the sequence, the hidden state  $\vec{h}^{(all)}(t)$ , which summarizes the whole sequence, is processed by a fully connected (FC) layer  $FC$  and Softmax function to obtain  $\vec{y}^{(all)} = [y_0^{(all)}, y_1^{(all)}]^T$ , with  $y_0^{(all)}$  and  $y_1^{(all)}$  representing the probability of the sequence state being negative or positive and  $y_0^{(all)} + y_1^{(all)} = 1$ . We refer to this conventional approach as *Basic GRU*. A schematic plot of the network is show in Fig. 2a. A compact representation of the neural network can be written as  $\vec{y}^{(all)} = f^{(all)}(\vec{x}^{(all)}; \mathbf{W}_{0,1}^{(all)})$ , with  $\mathbf{W}_0^{(all)}$  and  $\mathbf{W}_1^{(all)}$  being the network parameters that make decision for negative or positive predictions, respectively. It should be mentioned that  $\mathbf{W}_0^{(all)} \neq \mathbf{W}_1^{(all)}$  and  $\mathbf{W}_0^{(all)} \cap \mathbf{W}_1^{(all)} \neq \emptyset$ .

During our research, it was found out that feeding all signals into one neural network, however, has the consequence that the network over-fits to the dominant signal that provides the most information. As a result, we lose the information of other sensors that may have higher confidence about the driver's intentions in certain situations. Besides, if all signals are fed into one network, the network may over-fits to the correlations between sensors that do not generalize. To leverage the advantages of multi-sensors, we divide signal into  $n$  different categories (denoted as  $\vec{x}^{(k)}(t)$ ,  $k = 1, \dots, n$ ) and train for each category of sensors a individual network (denoted as  $f^{(k)}(x^{(k)}; \mathbf{W}_{0,1}^{(k)})$ ). At the end of a sequence, each network predicts the probability  $\vec{y}^{(k)} = [y_0^{(k)}, y_1^{(k)}]^T$  of hard brakes. The decision  $\vec{y}^* = [y_0^*, y_1^*]^T$  is made by focusing the sensor with the highest confidence for each state, computed as follows:

$$y_0^* = \max(y_0^{(1)}, \dots, y_0^{(n)}) \quad (1)$$

$$y_1^* = \max(y_1^{(1)}, \dots, y_1^{(n)}) \quad (2)$$

For example, a decision can be made as  $\vec{y}^* = [y_0^{(i)}, y_1^{(j)}]^T$ , meaning *sensor i* has the highest confidence for the sequence being negative while *sensor j* for positive with  $i \neq j$  (because  $y_0^{(k)} + y_1^{(k)} = 1, \forall k = 1, \dots, n$ ). We refer to the above described decision procedure as *RNN ensemble*. In addition to *RNN ensemble*, we further proposed *RNN focus* with modified parameters update scheme, stated as follows. The decision  $\vec{y}^*$  is compared with ground truth one-hot vector and the loss  $L$  is computed as cross-entropy; given this loss, network parameters are updated via back-propagation. It should be mentioned that *RNN focus* is different from training multiple networks on several signals and invoking vote mechanism. In our case, each network  $f^{(k)}(x^{(k)}; \mathbf{W}_s^{(k)})$  is only trained on a sample, if the network has the highest confidence for the prediction  $s$  among all networks. This can be better described in the update scheme as follows:

$$\mathbf{W}_0^{(i)} = -\eta \frac{\partial L}{\partial \mathbf{W}_0^{(i)}} \mathbb{1}_{i=\text{argmax}(y_0^{(1)}, \dots, y_0^{(n)})} + \mathbf{W}_0^{(i)} \quad (3)$$

$$\mathbf{W}_1^{(j)} = -\eta \frac{\partial L}{\partial \mathbf{W}_1^{(j)}} \mathbb{1}_{j=\text{argmax}(y_1^{(1)}, \dots, y_1^{(n)})} + \mathbf{W}_1^{(j)} \quad (4)$$

$$\mathbf{W}_{0,1}^{(k)} = \mathbf{W}_{0,1}^{(k)}, \quad \forall k \neq i \text{ or } j, \quad (5)$$

where  $\eta$  is learning rate and  $\mathbb{1}$  is indicator function. Unlike *Basic GRU*, where a  $f^{(all)}$  has to make decision for all scenarios, *RNN focus* forces each individual  $f^{(k)}$  to generate high probability for a state, only when it is very certain; otherwise, each  $f^{(k)}$  is allowed to be ambiguous (because of no update of parameters, described in Eq. 5). This operation can be done by Max-pooling. A schematic plot of the network is show in Fig. 2b.

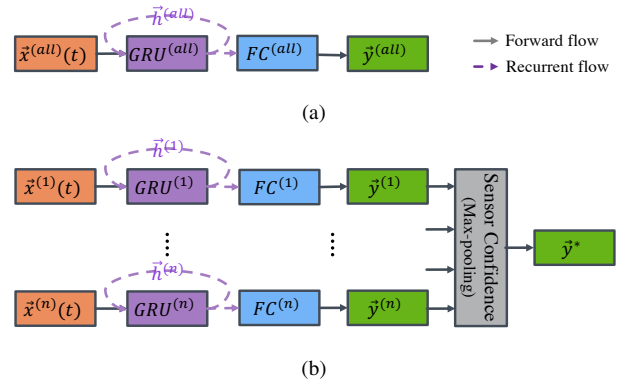


Fig. 2: (a) *Basic GRU*. (b) Proposed method *RNN focus* leveraging confidence of individual signal

The neural network implementation is done in Tensorflow. We use Relu for all activation functions and update parameters using RMSProp optimizer [33] with a learning rate of 0.0005. To prevent over-fitting, a dropout rate of 0.2 and L2 regularization with  $\lambda = 2 * 10^{-5}$  are applied. We train the network for 1500 steps where the network reaches convergence.

## V. EVALUATION AND RESULTS

### A. Unbalanced Dataset

Hard brakes are rare events and therefore we face the unbalanced dataset problem. To solve this problem, we collect negative samples as 5 times many as positive samples. During the training, we make sure the numbers of positive and negative samples are equal in every batch by random sampling. This procedure ensures that the neural network will not be over-fitted to the majority while being exposed to as many samples as possible. During the testing, we replicate positive test samples 5 times and therefore construct a balanced test dataset.

### B. Comparison between baseline, Basic GRU and proposed methods

An overview of the performance comparison is given in Tab. I. *Baseline* is the performance of Algorithm 1 using the optimal parameters. We run the experiments of *Basic GRU* with different categories of signals, where *Sub. Signals* means a subset of signals containing only accelerator pedal, speed of 4 wheels and steering wheel angle. The *RNN ensemble* differs from *RNN focus* in that *RNN ensemble* trains

each network individually (namely without using update scheme in Eq. 3-5) but makes decision based on Eq. 1-2.

We use one recurrent layer with 8 hidden units. Adding more recurrent layers or hidden units does not improve the performance while increases computation complexity, which is not desirable in time critical context.

|                  |                       | TPR    | FPR    | Accuracy      |
|------------------|-----------------------|--------|--------|---------------|
|                  | Baseline              | 64.50% | 41.01% | 61.75%        |
| <i>Basic GRU</i> | Accelerator pedal     | 63.99% | 17.79% | 73.10%        |
|                  | Speed of 4 wheels     | 57.05% | 20.86% | 68.09%        |
|                  | Long. and later. acc. | 45.54% | 25.63% | 59.96%        |
|                  | Steer. wheel angle    | 45.63% | 27.96% | 58.83%        |
|                  | All signals           | 57.81% | 17.99% | 69.91%        |
|                  | Sub. signals          | 60.16% | 18.30% | 70.93%        |
|                  | <i>RNN ensemble</i>   | 63.47% | 15.27% | 74.09%        |
|                  | <i>RNN focus</i>      | 65.83% | 14.90% | <b>75.47%</b> |

TABLE I: Performance Comparison

Compared with the baseline method, the *Basic GRU* with accelerator pedal improves the accuracy by 11.35%. This demonstrates that RNN has higher expressiveness of accelerator pedal movement than the baseline method. Signals from other sensors cannot achieve an accuracy as high as accelerator pedal. The *Basic GRU* using all signal as input does not provide a better result. The results show that fusing multiple sensors as inputs does not contribute to improvement, which is not inline with common sense.

The contradictory results indicate that the network may over-fit to relations between sensors that do not necessarily reflect a driver's intention. In addition, as shown in Tab. I, accelerator is the dominating signal for prediction. This means there is also risk that the neural network is train biased towards this signal. Especially, we found *Basic GRU* trained solely on accelerator, wheel speeds or steering wheel produce most divergent predictions, which means each category of signals captures a driver's intention from different perspectives. This leads to the choice of *Sub. Signals*. Targeting this problem, *RNN ensemble* and *RNN focus* device individual networks trained on these three categories of signals. The experiments show that we can benefit from such design. *RNN focus* outperforms *RNN ensemble* and improves *Basic GRU (Acc. pedal)* architecture by around another 2.37%.

Baseline, *Basic GRU* and *RNN focus* can be tuned to balance between true and false positive rate. For the baseline method, this is done by adjusting the *threshold* of accelerator pedal gradients in Algorithm 1. *Basic GRU* and *RNN focus* produce for a given input sequence the probability of the sequence being positive or negative. The default threshold value for decision making is 50%. To avoid high false positive rate, the threshold can increase and vice versa. The Receiver Operating Characteristic (ROC) and corresponding area under curve (AUC) are given in Fig. 3. As shown in the figure, for most given false positive rate, the *RNN focus* achieves a higher true positive rate than *Basic GRU (Acc. pedal)* and baseline.

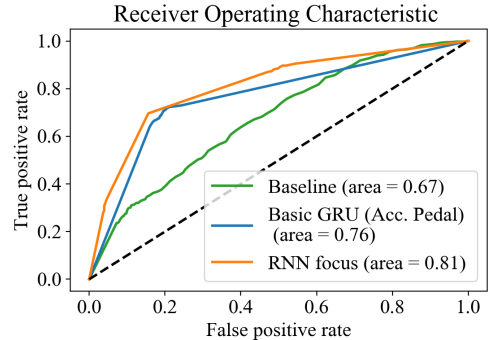


Fig. 3: Receive Operation Characteristic of both methods

### C. Insights into RNN decisions

Though neural networks are well-known for their unexplainable property, it is interesting to explore how and what features / patterns does the neural network use to differentiate itself from the baseline method. For this purpose, we plot several typical true positive / negative samples that are correctly predicted by *Basic GRU (Acc. Pedal)* and *RNN focus* but failed by the baseline method or *Basic GRU*, respectively. In both Fig. 4 and 5, we plot 5s of input signal and following 2s after the end of accelerator events. We omit longitudinal and lateral acceleration and plot only the average speed of four wheels and the steering wheel value because they alone provide enough and comparable insights for our explanations.

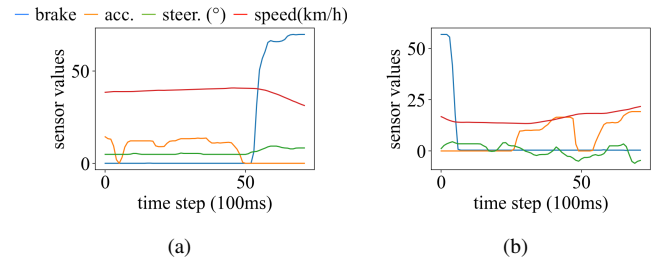


Fig. 4: True positive (a) and true negative (b) samples detected by *Basic GRU* but not by baseline method

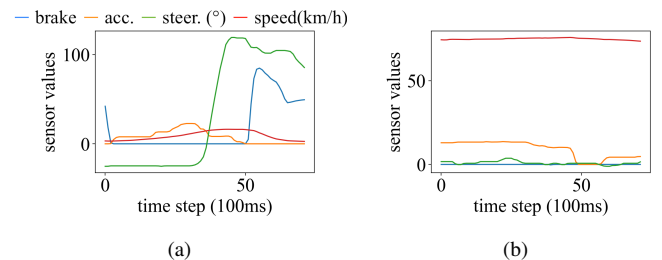


Fig. 5: True positive (a) and true negative (b) samples detected by the *RNN focus* but not by *Basic GRU*

We first compare *Basic GRU* and baseline method. As can be seen in Fig. 4a, the driver does not release the



accelerator pedal rapidly, which leads to a small gradient of the signal. However, the end of the accelerator event demonstrates an increased gradient (i.e. increased absolute value of first derivative), which means the driver releases the pedal in an accelerated way. This indicates that the driver has intention to decelerate. As a result, a brake maneuver follows. For negative samples, as shown in Fig. 4b, the driver accelerated moderately before he quickly released the accelerator pedal, which indicates that the release would not lead to successive deceleration. As a result, no further brake maneuver occurs. Though such movement patterns can also be defined mathematically, neural network approach offers a convenient way of modeling. The learning ability of neural networks further enables more flexible over-the-air adaptation to specific drivers.

Nevertheless, accelerator pedal signal alone can be misleading. We further compare *Basic GRU* and *RNN focus*. In Fig. 5a, the driver applies moderate force to the accelerator pedal before he fully releases it. However, an accelerator release accompanied with a large change in steering wheel angle indicates that the driver may just have made a turn into new lane and needs deceleration. In Fig. 5b, the release of accelerator pedal only has a medium large gradient, which leads to a mediocre probability of accelerator signal. Meanwhile, steering wheel and vehicle speed remain rather constant, indicating high probability of no brake maneuver. Summarizing the confidences of all sensors, *RNN focus* gives a negative prediction. A possible explanation for this example could be: the vehicle moves straightly at a high velocity; the driver has tiredness in her/his legs and therefore represses the accelerator pedal to relax.

#### D. Reliability of the RNN approach

To further exam the usefulness of the *RNN focus*, we test the trained the network *RNN focus* on a large balanced dataset that has relaxed condition on hard brakes, namely positive samples are accelerator events followed by lighter brake slam (15% instead of 25% of brake range within 0.1s) in 2s (instead of 1s). The performance in different risky scenario is plotted in Fig. 6. The more left, more upper positions of the matrix corresponds to more risky states. As can be seen, the more risky a scenario is, the better accuracy the *RNN focus* can achieve. Furthermore, in scenarios where brake pedal is pressed over 45% within 0.1s, the *RNN focus* achieves a true positive rate of 100% over the whole dataset. The *RNN focus* enables parallel processing of individual signals on separated small size neural network. An overview of computation time and model complexity are given in Tab. II. The arrangement of parallel architecture of the *RNN focus* allows us to not increase computation time dramatically while using more complex model as compared to *Basic GRU*. Nevertheless, neural network approaches are far more complicated than the baseline method, which can even be realized by analog circuits. Considering the importance of safety issue and the advancement of specialized neural processor, we believe the proposed method is promising for real applications. The computation time is evaluated for one

classification example on a CPU set-up with an Intel-i5 2.7 GHz and 8 GB 1867 MHz DDR3 RAM.

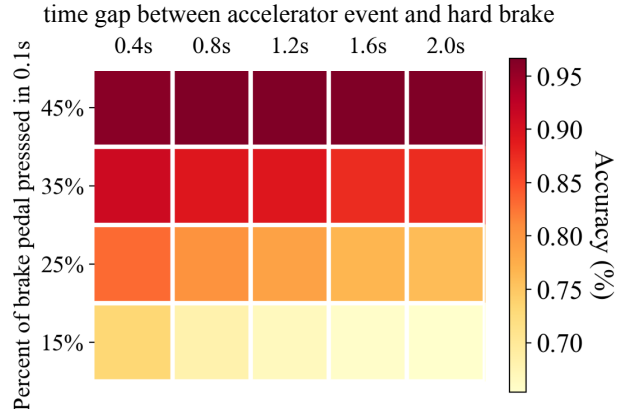


Fig. 6: Reliability of the *RNN focus*

|             | Baseline   | <i>Basic GRU</i><br>( <i>Acc. pedal</i> ) | <i>Basic GRU</i><br>( <i>All Signal</i> ) | <i>RNN focus</i> |
|-------------|------------|---|---|------------------|
| comp. time  | 42 $\mu$ s | 19.23 ms                                  | 20.74 ms                                  | 19.95 ms         |
| #parameters | 2          | 306                                       | 810                                       | 1134             |

TABLE II: Computation efficiency for one classification example

## VI. DISCUSSION AND OUTLOOK

This work targets a very practical yet important problem in industries. We showed that by leveraging RNN technique with focus on sensor confidence, prediction accuracy and AUC of ROC can be improved by around 13.72% and 14% as compared to the conventional algorithm. When compared to the neural networks without sensor confidence, we can improve the accuracy and AUC by around 2.37% and 5%. The proposed method does not require additional sensors in vehicles and enables convenient implementation in practices. While the advancement of artificial intelligence may lead eventually to autonomous vehicles (AV), the latest research shows that human drivers are 15 to 4000 times better than existing AV technologies for accidents per cumulative mile and drivers of AVs need to be as alert as drivers of non-AV vehicles [34]. Therefore, given the performance and the compatibility, we believe the proposed method can potentially increase driving safety by large margin in foreseeable future. Moreover, the result of this paper demonstrates that conventional sensor fusion approaches that use one neural network to process all CAN-Bus data can be sub-optimal for certain cases. This discovery suggests several potential directions for further exploration. On the one hand, the proposed *RNN focus* presents a novel way of sensor fusion, which can be applied to minimize uncertainty for intelligent transportation applications; on the other hand, more powerful regularization methods for neural networks are desirable to fully utilize heterogeneous sensors in modern vehicles. As such, we are excited for future works targeting depicted challenges.

## ACKNOWLEDGMENT

This work was supported by the Bosch IoT Lab, a collaboration of Bosch, ETH Zurich and the University of St. Gallen, Switzerland. The authors would also like to thank André Dahlinger, Benjamin Ryder and Yeyao Zhang for their valuable help and support.

## REFERENCES

- [1] R. Lozano, M. Naghavi, K. Foreman, S. Lim, K. Shibuya, V. Aboyans, and et al., "Global and regional mortality from 235 causes of death for 20 age groups in 1990 and 2010: a systematic analysis for the global burden of disease study 2010," in *The Lancet*, vol. 380, 2013, pp. 2095–2128.
- [2] A. Schieben, S. Griesche, T. Hesse, N. Fricke, and M. Baumann, "Evaluation of three different interaction designs for an automatic steering intervention," *Transportation Research Part F: Traffic Psychology and Behaviour*, pp. 238–251, Nov. 2014.
- [3] L. D. Adarns, P. Place, and L. D. Adarns, "Review of the literature on obstacle avoidance maneuvers: Braking versus steering," 1994.
- [4] S. K. J. S. Suzanne E Lee, Eddy Llaneras, "Analyses of rear-end crashes and near-crashes in the 100-car naturalistic driving study to support rear-signaling countermeasure development," *U.S. Department of Transportation, National Highway Traffic Safety Administration*, pp. 810–846, Oct. 2007.
- [5] M. Sieber, K.-H. Siedersberger, A. Siegel, and B. Färber, "Automatic emergency steering with distracted drivers: Effects of intervention design," in *2015 IEEE International Conference on Intelligent Transportation Systems, 2015, Las Palmas, Spain, Nov 15-18, 2015*. IEEE, 2015.
- [6] M. Heesen, M. Dziennus, T. Hesse, A. Schieben, C. Brunken, C. Löper, J. Kelsch, and M. Baumann, "Interaction design of automatic steering for collision avoidance: challenges and potentials of driver decoupling," in *IET Intelligent Transport Systems*, 2015, pp. 95–104.
- [7] J. Clauss, W. Klinkner, B. Knoff, and M. Steiner, "Verfahren zur durchführung eines automatischen bremsvorgangs," Patent EP0 819 597A1, Jan. 1998.
- [8] —, "Process for carrying out an automatic braking operation," Patent US5 924 508, Jul. 1999.
- [9] S. S. Ahn, "Methods and systems for automated control of vehicle braking," Patent US8 027 773, Sep. 2011.
- [10] E. OLOFSSON, "Improved accelerator pedal with braking action," Patent WO2 013 000 042A1, Jan. 2013.
- [11] M. Sieber and B. Färber, "Driver perception and reaction in collision avoidance: Implications for adas development and testing," *IEEE Intelligent Vehicles Symposium*, Jun. 2016.
- [12] B. JP and A. SA, "Alarm mistrust in automobiles: how collision alarm reliability affects driving," *Appl Ergon.*, pp. 34(6):499–509, 2003.
- [13] M. Saffarian, d. J. Winter, and J. Senders, "The effect of a short occlusion period on subsequent braking behavior: A driving simulator study," 2017. [Online]. Available: [https://www.researchgate.net/publication/314658202\\_The\\_effect\\_of\\_a\\_short\\_occlusion\\_period\\_on\\_subsequent\\_braking\\_behavior\\_A\\_driving\\_simulator\\_study](https://www.researchgate.net/publication/314658202_The_effect_of_a_short_occlusion_period_on_subsequent_braking_behavior_A_driving_simulator_study)
- [14] J. Lee, D. McGehee, T. Brown, and M. Reyes, "Collision warning timing, driver distraction, and driver response to imminent rear-end collisions in a high-fidelity driving simulator," *Hum Factors*, pp. 44(2):314–34, 2002.
- [15] K. Fuhrmann, D. Peters, W. Kreft, and D. Baumgarten, "Bremsvorwarnsystem," Patent EP0 890 477A2, Jan. 1999.
- [16] A. Taylor, S. Leblanc, and N. Japkowicz, "Anomaly detection in automobile control network data with long short-term memory networks," in *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, 2016.
- [17] D. Hallac, S. Bhooshan, M. Chen, K. Abida, R. Sosič, and J. Leskovec, "Drive2Vec: Multiscale State-Space Embedding of Vehicular Sensor Data," in *2018 IEEE 21th International Conference on Intelligent Transportation Systems (ITSC)*, 2018.
- [18] M. M. Trivedi, T. Gandhi, and J. C. McCall, "Looking-in and looking-out of a vehicle: Computer-vision-based enhanced vehicle safety," *IEEE Transactions on Intelligent Transportation Systems*, pp. 374–387, Feb. 2007.
- [19] A. Jain, H. S. Koppula, S. Soh, B. Raghavan, A. Singh, and A. Saxena, "Brain4cars: Car that knows before you do via sensory-fusion deep learning architecture," *CoRR*, vol. abs/1601.00740, 2016. [Online]. Available: <http://arxiv.org/abs/1601.00740>
- [20] J. C. McCall and M. M. Trivedi, "Driver behavior and situation aware brake assistance for intelligent vehicles," *Proceedings of the IEEE*, pp. 374–387, Feb. 2007.
- [21] C. Tran, A. Doshi, and M. M. Trivedi, "Pedal error prediction by driver foot gesture analysis: A vision-based inquiry," *IEEE Intelligent Vehicles Symposium*, 2011.
- [22] —, "Modeling and prediction of driver behavior by foot gesture analysis," *Comput. Vis. Image Underst.*, vol. 116, no. 3, pp. 435–445, Mar. 2012. [Online]. Available: <http://dx.doi.org/10.1016/j.cviu.2011.09.008>
- [23] D. Feng, L. Rosenbaum, F. Timm, and K. Dietmayer, "Leveraging heteroscedastic aleatoric uncertainties for robust real-time lidar 3d object detection," *arXiv preprint arXiv:1809.05590*, 2018.
- [24] D. Feng, L. Rosenbaum, and K. Dietmayer, "Towards safe autonomous driving: Capture uncertainty in the deep neural network for lidar 3d vehicle detection," in *2018 IEEE 21th International Conference on Intelligent Transportation Systems (ITSC)*, Nov. 2018, pp. 3266–3273.
- [25] E. Ilg, O. Çiçek, S. Galesso, A. Klein, O. Makansi, F. Hutter, and T. Brox, "Uncertainty estimates and multi-hypotheses networks for optical flow," 2018.
- [26] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?" in *Advances in Neural Information Processing Systems*, 2017, pp. 5574–5584.
- [27] N. Ueda and R. Nakano, "Generalization error of ensemble estimators," in *1996 IEEE International Conference on Neural Networks*, 1996.
- [28] F. Barata, K. Kipfer, M. Weber, P. Tinschert, E. Fleisch, and T. Kowatsch, "Towards device-agnostic mobile cough detection with convolutional neural networks," in *2019 IEEE 7th IEEE International Conference on Healthcare Informatics*, 2019.
- [29] X. Chen, A. S. Vempati, and P. Beardsley, "Streetmap - mapping and localization on ground planes using a downward facing camera," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.
- [30] B. Gahr, B. Ryder, A. Dahlinger, and F. Wortmann, "Driver Identification via Brake Pedal Signals A Replication and Advancement of Existing Techniques," in *2018 IEEE 21th International Conference on Intelligent Transportation Systems (ITSC)*, 2018.
- [31] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, vol. abs/1412.3555, 2014.
- [32] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
- [33] T. Tieleman and G. Hinton, "Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude," COURSERA: Neural Networks for Machine Learning, 2012.
- [34] S. S. Banerjee, S. Jha, J. Cyriac, Z. T. Kalbarczyk, and R. K. Iyer, "Hands off the wheel in autonomous vehicles?: A systems perspective on over a million miles of field data," in *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Jul. 2018.