Lane Change Prediction With an Echo State Network and Recurrent Neural Network in the Urban Area

Karoline Griesbach, Matthias Beggiato, and Karl Heinz Hoffmann

***Abstract*— The prediction of lane changes can reduce traffic accidents and improve traffic flow. In this paper two classifiers, Echo State Network and a recurrent neural network with Long Short Term Memory cells, were compared to predict lane changes using the input variables steering angle and indicator. The input variables were extracted from a data set which was generated from a naturalistic driving study in the urban area of Chemnitz, Germany. Both classifiers predicted left and right lane changes successfully. They achieved high true positive rates and low false positive rates. The Echo State Network predicted left and the recurrent neural network predicted right lane changes better.**

***Index Terms*— Echo state network, recurrent neural network, LSTM, lane change prediction, urban area.**

I. Introduction

AN ESSENTIAL research field in the area of vehicle development is the prediction of driver behavior, such as turning at intersections or lane changes. A wide range of research focused on lane change prediction on highways. Nowadays, research in this area focuses lane change prediction in urban areas as well. The prediction of lane changes in urban areas is more complicated than on highways. Lane change executions in urban areas are affected by complex traffic rules and various traffic participants, such as pedestrians, cyclist, cars, and public transportation.

The development of a successful lane change prediction classifier and its implementation in the vehicle as an advanced driver assistance system (ADAS) can reduce traffic accidents and traffic congestions [1]–[3]. A common reason for traffic accidents are drivers' misbehavior during lane changes [4]. Integrating the lane change prediction classifier in the ADAS can reduce these accidents. The ADAS will warn the driver of unsafe maneuvers and activate helpful and deactivate interfering ADAS to avoid conflicting or annoying situations.

Moreover, a lane change prediction classifier can improve the traffic flow through vehicle-to-vehicle communication in the future. The lane change prediction classifier warns surrounding vehicles about upcoming lane changes. Thus, sudden braking, which can lead to traffic congestions or accidents, is avoided, because the vehicle/driver can react in time. Particularly the mixed traffic situation of autonomous and non-autonomous vehicles will benefit from such an implementation.

In this paper, an Echo State Network (ESN) and a recurrent neural network (RNN) are introduced to predict lane changes in the urban area of Chemnitz, Germany. The lane change prediction with the ESN and RNN needed fewer input variables and used more previous time points of each input variable compared to previous studies. The input variable indicator and steering angle have the advantage that they can be easily accessed via the ego vehicle's CAN-bus-Logger.

The next section summarizes existing methods for lane change prediction. Then, the lane change prediction algorithm is introduced. Finally, the classifiers were compared with each other.

II. Previous Research

To predict lane changes variables are needed describing the characteristics of lane changes. These variables are divided usually into three categories: driver attributes (e.g., gazes), vehicle attributes (e.g., steering angle), and environment attributes (e.g., distance to other vehicles). A comprehensive summary is given in [5].

In the past, different classifiers from the filed machine learning were applied to predict lane changes, for example, hidden Markov models (HMMs), Bayesian networks, or neural networks. In this paper, existing methods are introduced which use naturalistic driving data to predict left lane changes. Two kinds of existing methods are analyzed: studies using the same data set as in this paper and studies which performed well using other data sets. A summary of past research is given in [5].

The analysis of the existing methods includes the input variables, the preprocessing of the input variables, and the performance rate of the classifiers. Furthermore, only classifiers are considered using data from naturalistic driving studies.

A. Studies Using the Same Data Set

One approach used a neural network and data fusion to predict left and right lane changes [6]. The neural network's

Manuscript received May 11, 2020; revised November 8, 2020 and January 30, 2021; accepted February 1, 2021. The Associate Editor for this article was H. Dia. *(Corresponding author: Karoline Griesbach.)*

Karoline Griesbach and Karl Heinz Hoffmann are with the Theoretical Physics, especially Computational Physics, TU Chemnitz, 09126 Chemnitz, Germany (e-mail: karolinegriesbach@gmx.de; hoffmann@physik.tu-chemnitz.de).

Matthias Beggiato is with the Cognitive and Engineering Psychology, TU Chemnitz, 09107 Chemnitz, Germany (e-mail: matthias.beggiato@psychologie.tu-chemnitz.de).

Digital Object Identifier 10.1109/TITS.2021.3058035

input contained three subsets: driver attributes (head's three-dimensional position and orientation in the directions of the axis x, y, and z), environment attributes (accessibility of a lane), and vehicle attributes (lateral distance to the closest lane marking, velocity, yaw rate, brake and gas pedal pressure of the ego vehicle). In total seven variations of an input variable were considered: the current considered input variable, the minimum and maximum of the previous 0 to 2 s, 2 to 4 s, and 4 to 10 s to a lane change were used as input. Thus, every input variable had seven input neurons. Each of the subsets had one output which was forwarded as new input to a hidden layer with three neurons. The final output was a single neuron which returned the status lane change or no lane change. The input variables were obtained from a naturalistic driving study in the urban area of Chemnitz, Germany. The neural network predicted 98.3% of the left lane changes 2 s before their occurrence correctly. The false alarm rate was 0.038. The accuracy (94.2%) decreased and the false alarm rate (0.063) increased 4 s prior to left lane changes [6].

In a subsequent study [7], the same network structure was used, but the lateral distance to the close lane marking was removed from the input variables. This network achieved an accuracy (*ACC*) of 99.1% and a false alarm rate of 0.018 2 s before a lane change. This neural network predicted better left lane changes than the above one [6].

Furthermore, a fuzzy pattern classification was applied for left and right lane change prediction using the same data set [8]. Only driver attributes were used as input: cumulated gazes to the left, right and rearview mirror for 8 s intervals. The classifier could predict lane changes approximately 7 s before they occurred. The *TPR* was 81.6% for left and 79.3% for right lane changes. The *TPR* values were increased by applying the correlated *TPR* (left: 89.3%, right: 86.3%) which was calculated by correcting misclassifications by a transition pattern classifier in the first 0.5 s [8].

B. Other Studies

A HMM was used to predict left and right lane changes of a lead vehicle on straight and curved roads [9]. The distance between the ego and lead vehicle, as well as the lateral and longitudinal velocity of the lead vehicle were the inputs of the HMM. The input values contained the current and previous data points (4.5 s for straight and 3.5 s for curved roads). The input values were collected with a naturalistic driving study. The HMM predicted lane changes on straight roads 1.2 s after the lane change started and with an accuracy of 97%. For curved roads, it predicted lane changes 1 s after they started with an accuracy of 96% [9].

Another study compared a dynamic Bayesian network (DBN) to a predictive control based approach, a Bayesian network, a RNN, a HMM, and a rule-based approach for left and right lane change prediction on highways [10]. The input contained vehicle and environment attributes: indicator activation, activation of the brake, the direction of lateral velocity, the yaw rate to the road tangent, the existence of left or right adjacent lanes, the existence and relative velocity between ego vehicle and rear or front vehicle in the adjacent

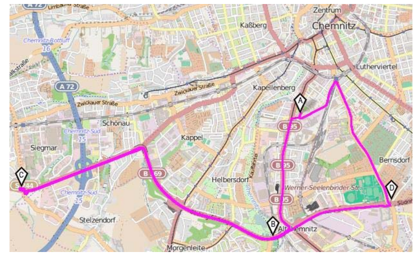


Fig. 1. The driving route in Chemnitz, Germany, of the naturalistic driving study (OpenStreetMap contributors, 2016).

left and right lane, the existence and relative velocity to a lead vehicle in the same lane, the classification of a vehicle (e.g., motorcycle or truck), and the distance to a neighboring lane from the ego vehicle. The input variables at the current time point and 1 s previous of the current time point were used as input. In total, 753 lane changes were extracted of the data sets NGSIM I-80 and US-101. The DBN achieved the best F1-Score (0.83), a lower accuracy (0.72) than the predictive control based approach (0.81), RNN (0.89) and HMM (0.91). The DBN achieved the second highest prediction time of 3.75 s. The predictive control based approach obtained the best one (4.56 s) [10].

C. Summary

The HMM and DBN used vehicle and environment attributes as inputs. The neural network had as additional input driver attributes. The fuzzy approach predicted left and right lane changes using only driver attributes as input. In general, a combination of different attributes led to good prediction accuracies.

All studies used current and previous time points of the input variables, thus considering both seem to achieve better predictions than using only current time points.

III. Method

The prediction of left and right lane changes was realized with the input variables steering angle and indicator. Various input combinations were tested, and this one was the most successful one. A short overview is given in section III-C.

The lane change prediction is realized with an ESN and compared to a RNN. Both networks are very suitable for time series prediction. Furthermore, the RNN was chosen for comparison because it can be better compared to existing studies which used the same data set [6], [7]. In the following, the data set and parameter settings of the two classifiers are explained.

A. Data Set

The data set was provided by [11]. The considered data set contained 57 test persons who drove a 40 km long route in the urban area of Chemnitz, Germany (Fig. 1). They drove a VW Touran with automatic transmission which was equipped with a data logging system (CAN-bus). Various driver (e.g. age, gaze direction), vehicle (e.g. velocity, steering angle), and environment attributes (e.g. distance to lead vehicle) were

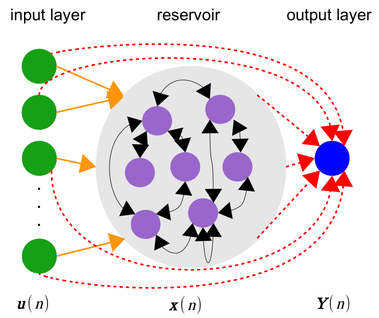


Fig. 2. Scheme of ESN with the input ($\mathbf{u}(n)$) at time point n, reservoir states ($\mathbf{x}(n)$), output ($\mathbf{Y}(n)$), input weights ($\mathbf{W}^{\text{in}}$), recurrent weights ($\mathbf{W}$) and output weights ($\mathbf{W}^{\text{out}}$). The connections with the dashed line have to be trained.

recorded. The environment attributes were extracted through radar sensor and video recordings. The lane changes were extracted through video recordings of the vehicle environment [12]. The lane change occurrence was annotated when the middle of the vehicle crossed the line between two lanes. In the end, seven lane change types were extracted: obstacle, slow lead vehicle, turn-off lanes, entering, merging vehicle, return after overtaking, and other unknown reasons.

Finally, 1748 lane changes were extracted (1019 left and 729 right lane changes).

B. Introduction to the Echo State Network

A supervised learning algorithm which is suitable for temporal data processing like time series prediction is the ESN [13]. [14]. The ESN was implemented according to the descriptions in [15], [16].

A scheme of the ESN is illustrated in Fig. 2. The input neurons are displayed in the input layer. Their data are saved as a vector $\mathbf{u}(n) \in \mathbb{R}^{N_\text{u}}$, where $n = 1 \ldots D$ is the considered time point and N_u is the total number of input neurons.

The inputs are connected with each reservoir neuron in the reservoir. The matrix $\mathbf{W}^{\text{in}} \in \mathbb{R}^{N_\text{x} \times N_\text{u}}$ contains the connections, N_x is the total number of reservoir neurons. The reservoir neurons activations' are saved in the vector $\mathbf{x}(n)$ for each time point n. The reservoir neurons are connected with each other. These connection are saved in the recurrent matrix $\mathbf{W} \in \mathbb{R}^{N_\text{x} \times N_\text{x}}$. The output neurons are connected with the reservoir and input neurons by the output matrix $\mathbf{W}^{\text{out}} \in \mathbb{R}^{N_\text{y} \times (N_\text{x}+N_\text{u})}$, where N_y is the total number of output neurons. The output of the ESN is the matrix $\mathbf{Y} \in \mathbb{R}^{D \times N_\text{y}}$. In general ESNs can have feedback connections from the output to the reservoir neurons which are not applied in this work.

Several steps have to be considered to implement an ESN given an input sequence:

1. create a large random reservoir (initialization of the recurrent weight matrix)
2. calculate the activation of the reservoir neurons for each time step
3. calculate the output weight matrix (training), and
4. test the ESN with data not used in training.

1) Create a Large Random Reservoir: First, the reservoir size N_x has to be defined. To avoid overfitting, the number of reservoir neurons should not be too big. According to [16], a reservoir is too big when not enough training data D is available $D < N_\text{u} + N_\text{x}$.

Then, random values from a uniform or normal distribution are generated to initialize the input and reservoir matrices. In addition, the echo state property was enforced, see [16].

2) Calculate the Activation of the Reservoir Neurons: The reservoir neurons $\tilde{\mathbf{x}}(n)$ and the activation of the reservoir neurons $\mathbf{x}(n)$ are updated for each time point n with:

$$\tilde{\mathbf{x}}(n) = \tanh(\mathbf{W}^{\text{in}}\mathbf{u}(n) + \mathbf{W}\mathbf{x}(n-1)). \tag{1}$$

The update is determined by the input matrix $\mathbf{W}^{\text{in}}$, current input $\mathbf{u}(n)$, the recurrent weight matrix $\mathbf{W}$, and the previous reservoir activation $\mathbf{x}(n-1)$. The reservoir activation is initialized with 0, thus $\mathbf{x}(n) = [0, \ldots, 0]$. While the most popular activation function is $tanh$, which was used here, other sigmoid functions can be used as well.

The new reservoir activation is determined by the leaking rate α, and is a combination of the previous activation state and the reservoir update $\tilde{\mathbf{x}}(n)$:

$$\mathbf{x}(n) = (1-\alpha)\mathbf{x}(n-1) + \alpha\tilde{\mathbf{x}}(n). \tag{2}$$

The reservoirs' update is regulated by the leaking rate.

3) Calculate the Output and Output Weight Matrix: The state matrix $\mathbf{X} \in \mathbb{R}^{D \times (N_\text{u}+N_\text{x})}$ which contains the inputs and reservoir activations ($\mathbf{X} = [\mathbf{u}(n); \mathbf{x}(n), \ldots, \mathbf{u}(D); \mathbf{x}(D)]$) and the output matrix $\mathbf{W}^{\text{out}}$ determine the ESN output. The output is then determined as:

$$\mathbf{Y} = \mathbf{W}^{\text{out}}\mathbf{X}. \tag{3}$$

In most cases, the state matrix X is used after an initial time lag to reduce initial effects.

A well-established approach in neural network training is the adaptation of all weights between all layers: input layer, hidden layers, and output layer. The ESN's advantage is that only the output weights $\mathbf{W}^{\text{out}}$ have to be trained. The calculation of the output weights is based on ridge regression [16]:

$$\mathbf{W}^{\text{out}} = \mathbf{Y}^{\text{target}}\mathbf{X}^{\text{T}}(\mathbf{X}\mathbf{X}^{\text{T}} + \beta\mathbf{I})^{-1}. \tag{4}$$

To find the best fit of $\mathbf{W}^{\text{out}}$ regarding the target output, the state matrix, regularization coefficient β, and the identity matrix $\mathbf{I} \in \mathbb{R}^{D \times D}$ ridge regression is used. To guarantee invertibility the regularization coefficient β is used. According to [16] $\beta = 1e^{-8}$ is chosen for further calculations.

C. Selection of Relevant Input Variables

In a previous study, eleven input combinations were tested to predict left and right lane changes with an ESN. The best combination included indicator, steering angle, and gaze [5]. For this paper, another input combination, steering angle and indicator, was added. The variables steering angle and indicator are easier to measure than gaze. Table I shows the results for the ESN predicting lane changes with the different input combinations using the metric area under the curve (AUC). The input combinations included variables from driver, environment, and vehicle attributes. Surprisingly, input combination 11 shows the best value of the AUC and was therefore chosen as input combination to predict lane changes with the ESN and RNN.

TABLE I
THE AUC VALUES WHICH THE ESN ACHIEVED PREDICTING LANE CHANGES WITH DIFFERENT INPUT COMBINATIONS

Number	Variable combination	AUC
0	velocity, brake pressure, gas pedal	0.78
1	velocity, indicator, gaze	0.92
2	velocity, brake pressure, gaze	0.92
3	steering angle, indicator, head rotation in y-direction	0.64
4	existence of lead vehicle, gas pedal, head pose	0.76
5	gas pedal, brake pressure, head rotation in y-direction	0.75
6	velocity, brake pressure, gas pedal, gaze	0.91
7	velocity, brake pressure, gas pedal, head rotation in y-direction	0.78
8	steering angle, indicator, gaze	0.91
9	velocity, gaze	0.92
10	existence of lead vehicle, gaze	0.89
11	indicator and steering angle	0.93

TABLE II
TRAINING AND TEST DATA FOR THE ESN PREDICTING LEFT AND RIGHT LANE CHANGES. FIRST NUMBER ARE THE LEFT LANE CHANGES, SECOND NUMBER ARE RIGHT LANE CHANGES

Lane change type	ESN 1st 33%	ESN 2nd 33%	ESN 3rd 33%
turn-offs	76\75	75\75	77\78
merging	73\0	74\0	78\0
unknown	11\0	19\5	26\1
entering	11\0	10\0	10\0
slow lead vehicle	147\0	162\0	162\0
return	0\151	0\172	0\172
obstacle	2\0	4\0	2\0
total lane changes	320\226	344\252	355\251

D. Lane Change Prediction With the Echo State Network

For lane change prediction the ESN was initialized with the following parameters:

- input variables steering angle and indicator
- input variables were scaled with standardization
- number of input neurons 300 (100 for each input variable, because the previous 99 data points were considered as well)
- learning rate $\alpha = 0.3$
- regularization coefficient $\beta = 1e^{-8}$
- initial time lag of 100
- number of reservoir neurons $N_{\mathrm{x}} = 4500$
- the input weights $\mathbf{W}^{\mathrm{in}}$ and recurrent weights $\mathbf{W}$ were initialized from a uniform distribution with the range [-0.3,0.3).
- 10-fold cross-validation

The data set was divided into three groups: the first, second, and third 33% of the total data set. The data set had in total n entries, so the first part contained the entries 1 till $n * 0.33$, the second part contained the part $n * 0.33 + 1$ till $n * 0.66$ and the third part $n * 0.66 + 1$ till $n * 0.99$. The details are shown in Table II. The advantage of dividing the data set in three subsets is the possibility to test the ESN's robustness during training. If the training and testing of the different data sets show similar results, it can be assumed that the ESN generalized.

The target output was set to 1 or -1 five seconds prior to left or right lane changes, respectively. The rest, the no lane changes, were set to 0.

After training the ROC curves for each dataset and cross-validation were calculated. To test the model, the trained output weights are required. The output weights of the best cross-validation of all data sets were used. The best one had a high true positive rate (*TPR*), a low false positive rate (*TPR*) and a high AUC value. With these output weights, lane change prediction of randomly unknown data which was not used for training was calculated.

The calculated output $\mathbf{Y}$ of the data was post-processed with a moving sum over a window length $p = 30$ data points:

$$y_{\mathrm{sum}}(n) = \sum_{s=n-p}^{n} y(s), \text{ if } s < 1, \text{ then } s = 1, \tag{5}$$

with n describing the current time point of the ESN output y. The advantage of this post-processing is that lane change values get emphasized and no lane change values weakened. This output is saved in the matrix $\mathbf{Y}_{\mathrm{sum}}$ for each output neuron and each time point of the test data. Afterwards, the threshold function is applied on $\mathbf{Y}_{\mathrm{sum}}$:

$$\mathbf{Y}_{\mathrm{pred}} = \begin{cases} 1 & \text{left lane change if } \mathbf{Y}_{\mathrm{sum}} \geq 20 \\ -1 & \text{right lane change if } \mathbf{Y}_{\mathrm{sum}} \leq -11 \\ 0 & \text{else} \end{cases} \tag{6}$$

Thus, left lane changes were classified if $\mathbf{Y}_{\mathrm{sum}} \geq 20$ and right lane changes were classified if $\mathbf{Y}_{\mathrm{sum}} \leq -11$. Due to the target coding of 1 for left lane changes and -1 for right lane changes, the output adapted to positive and negative values, respectively. Various thresholds were compared with each other. The thresholds 20 and -11 (see (6)) returned the best prediction results which means high *TPR* and low *FPR* for left and right lane change prediction. The variations of thresholds and the according results are displayed in Fig. 3.

Due to the target coding of 1 for left lane changes and -1 for right lane changes, the output adapted to positive and negative values, respectively.

With the output $\mathbf{Y}_{\mathrm{pred}}$ a ADAS would warn the driver several times before a lane change. This probably would annoy the drivers. The goal is to realize a single lane change prediction as soon as possible. For this reason, $\mathbf{Y}_{\mathrm{final}}$ is introduced.

If $\mathbf{Y}_{\mathrm{pred}} = 1$ occurred 12 times in a row, then $\mathbf{Y}_{\mathrm{final}}$ was set to 1 and predicted a left lane change. If $\mathbf{Y}_{\mathrm{pred}} = -1$ occurred six times in a row, then $\mathbf{Y}_{\mathrm{final}}$ was set to -1 and predicted a right lane change. A variation of values were tested to find the here mentioned best values for lane change prediction.

E. Recurrent Neural Network

RNNs were successfully used for lane change prediction, especially a RNN with Long Short Term Memory (LSTM) cells [17]. Here, a similar approach was applied. The RNN was

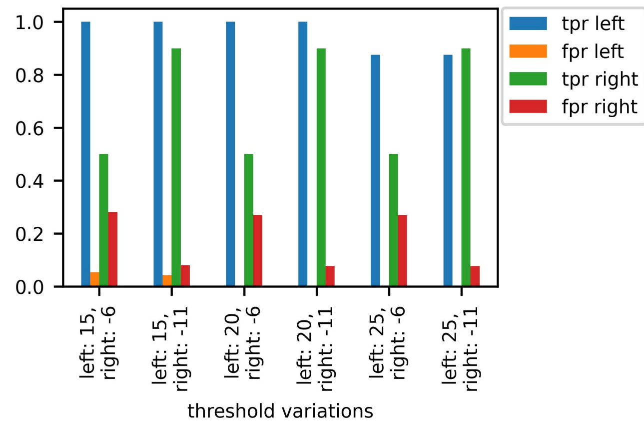


Fig. 3. The TPR and FPR for predicting left and right lane changes with $\mathbf{Y}_{\text{pred}}$ dependent on different thresholds. High TPR and low FPR indicate good performances.

initialized with 128 hidden neurons (LSTM cells) as in [17]. A detailed explanation of LSTM cells is given in [18]. In this case, just one output neuron for both lane change directions was considered. So it was comparable to the ESN. The scheme of the RNN is shown in Fig. 4 and was implemented with tensorflow [19].

The input variables were identical with the ESN input. The RNN was initialized with the following parameters:

- input variables steering angle and indicator
- input variables were scaled with standardization
- number of input neurons 300 (100 for each input variable, because the previous 99 data points were considered as well)
- RNN containing 128 LSTM cells
- training algorithm: Adam optimizer with the learning rate $\alpha = 0.001$ [20]
- training iterations: 100.
- 10-fold cross-validation

The data subsets and target output was the same as for the ESN.

As for the ESN the ROC curves were calculated and the trained weights which achieved the best performance were selected. The moving sum over a window length of 30 data points $\mathbf{Y}_{\text{sum}}$ was calculated as well. The threshold function to calculate $\mathbf{Y}_{\text{pred}}$ was adapted for the RNN as following:

$$\mathbf{Y}_{\text{pred}} = \begin{cases} 1 & \text{left lane change if } \mathbf{Y}_{\text{sum}} \geq 15 \\ -1 & \text{right lane change if } \mathbf{Y}_{\text{sum}} \leq -20 \\ 0 & \text{else} \end{cases} \tag{7}$$

As for the ESN the ROC curves were calculated and the trained weights which achieved the best performance were selected and tested on unknown data. The moving sum over a window length of 30 data points $\mathbf{Y}_{\text{sum}}$ was calculated as well. The threshold function to calculate $\mathbf{Y}_{\text{pred}}$ was adapted for the RNN as following.

To realize a final, single prediction $\mathbf{Y}_{\text{final}}$ for the RNN different values were tested, too. If $\mathbf{Y}_{\text{pred}} = 1$ occurred 12 times in a row, a left lane change was predicted. A right lane change was predicted immediately.

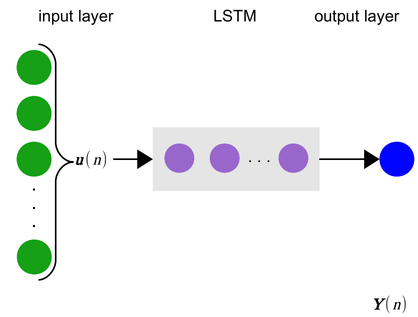


Fig. 4. Scheme of the RNN with the input vector $u(n)$ at time point n. The square represents the LSTM cell and it's neurons. The RNN has one output neuron with activation $\mathbf{Y}(n)$.

TABLE III
RESULTS OF THE ESN AND RNN FOR $\mathbf{Y}_{\text{final}}$

	TPR in %	FPR in %	ACC in %	$F1$
ESN left	100	0	100	1
RNN left	100	7.7	94	0.95
ESN right	90	7.9	92	0.85
RNN right	100	0	100	1

F. Results

The ROC curves and the means and standards deviation of the area under the curve (AUC) over all data sets and cross-validations were calculated. According to the ROC curves the ESN (Fig. 5a, 5c) predicted left and right lane changes better than the RNN (Fig. 5b, 5d).

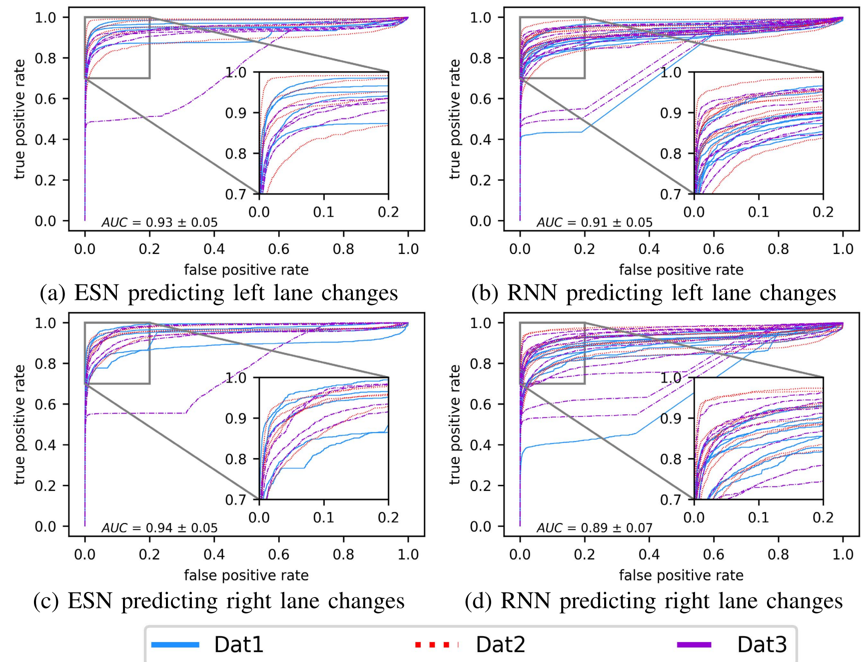
Table III shows the *TPR*, *TPR*, accuracy (*ACC*) and *F1* for left and right lane change prediction using the unknow data for both classifiers. The ESN predicted left lane changes better than the RNN, because it achieved a higher *TPR*, *ACC*, *F1*, and a lower *TPR*. The RNN predicted right lane changes better than the ESN, because it has a perfect classification with $TPR = 100\%$ and $FPR = 0\%$. Thus, all right lane changes were predicted correctly and no false alarms occurred.

The output of the ESN is displayed in Fig. 6a. The above plot shows the target, the second the calculated ESN $\mathbf{Y}$, the third $\mathbf{Y}_{\text{sum}}$ and the fourth the predicted output $\mathbf{Y}_{\text{pred}}$, the final single output $\mathbf{Y}_{\text{final}}$ and $\mathbf{Y}_{\text{wrong}}$ marks the false positives and negatives.

Two false positives occurred. One due to the fact that a right lane change was predicted right after the occurrence of a right lane change. The other was predicted 5.066 s before a right lane change. The targets for right lane changes was -1 which was set 5 s before a right lane change occurrence. So the ESN predicted it more than 5 s before its occurrence. For this lane change, a false negative was afterward detected.

Fig. 6b shows the output of the RNN. Here, two false alarms for left lane changes occurred, due to the fact that the inputs were activated, but no lane change maneuver was performed.

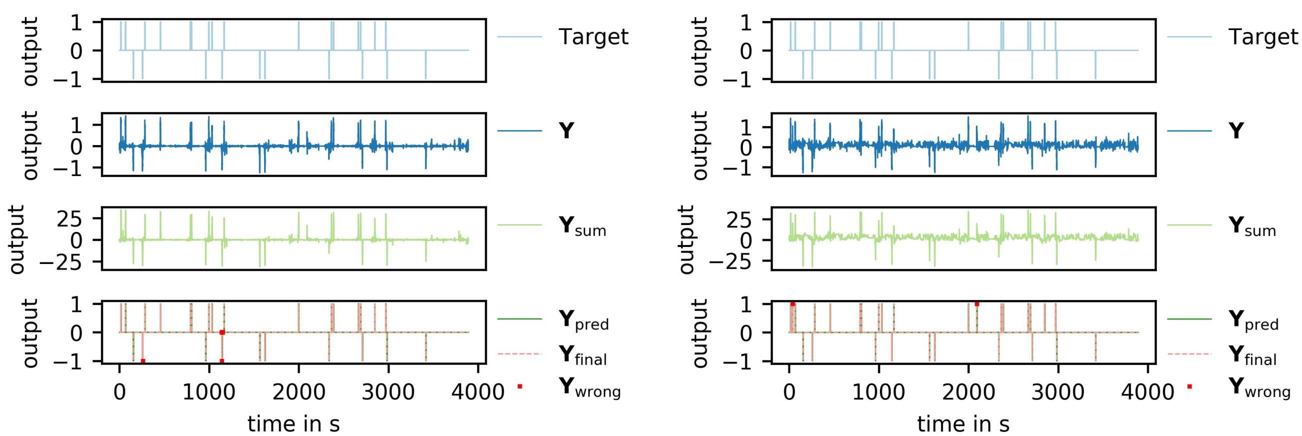
In average the ESN could predict left lane changes 2.84 s ($SD = 0.31$ s) before its occurrence and right lane changes with 4.07 s ($SD = 0.8$ s). The RNN predicted left lane changes 2.88 s ($SD = 0.68$ s) before its occurrence and right lane changes with 3.3 s ($SD = 0.63$ s). The ESN achieved a higher prediction time for right lane changes. For left lane change,



(a) ESN predicting left lane changes (b) RNN predicting left lane changes

(c) ESN predicting right lane changes (d) RNN predicting right lane changes

Fig. 5. ROC curves of the ESN and RNN for left and right lane change prediction. Each curve belongs to one cross-validation and one data set which results in 30 curves for each network and lane change direction. All zoom frames show that the lines are generally in the right upper corner of the figure which indicates good lane change prediction because high AUC values, high TPR and low FPR were achieved. The ESN has a lower average AUC value for left lane change prediction than for right lane change prediction. For the RNN the average AUC value for right lane changes is lower than for left lane changes. The RNN has three curves for left and right lane change prediction which show low true positive rates and high false positive rates at the beginning.



(a) ESN (b) RNN

Fig. 6. Output of the ESN and RNN for left and right lane change prediction. The upper subplot shows the target, the second plot the output using the moving sum $\mathbf{Y}_{\text{sum}}$, the third the predicted output $\mathbf{Y}_{\text{pred}}$, the predicted single output $\mathbf{Y}_{\text{final}}$ and the false positives and false negatives $\mathbf{Y}_{\text{wrong}}$. Note the few false positives and false negatives for the ESN and RNN.

the prediction time for ESN and RNN are comparable, because the mean has only a difference of 0.04 s and the standard deviation of the RNN is higher.

IV. Conclusion

The comparison of an ESN and RNN for lane change prediction revealed that none of them outperformed the other one. Both achieved similar prediction results. It is interesting that the ESN predicts left and the RNN predicts right lane changes better regarding different criteria.

Furthermore, the ESN and RNN predicted left and right lane changes using the two input variables steering angle and indicator with an accuracy over 90%.

Compared to the study of [8], the ESN and LSTM achieved higher *TPR*, but a lower prediction time. They achieved an prediction time of 7 s, but could only predict lane changes which included glances. Nevertheless, the prediction time of 2.2. to 4 s is sufficient to warn a driver. An anticipation time of at least 2 s is suggested for lane change assistance systems [21].

The neural network of [7] predicted only left lane changes with an $ACC = 99.1\%$ 2 s before its occurrence. Evaluating the accuracy for the ESN and RNN 2 s before a left lane change occurrence an $ACC = 1$ for both classifiers is achieved. A further advantage is that fewer input variables were used, but more previous time points of each input variable were considered compared to [7]. In this paper, left and right lane changes were predicted with one single ESN or RNN, whereas [7] used two different neural networks for each lane change direction.

Compared to the study of [9] the ESN achieved higher TPR for predicting left lane changes and lower TPR for predicting right lane changes. The RNN is for both directions better. Compared to the study of [10] the ESN and RNN achieved higher TPR for predicting left and right lane changes.

All in all, the ESN and LSTM are promising classifiers to predict left and right lane changes.

REFERENCES

[1] J. A. Laval and C. F. Daganzo, "Lane-changing in traffic streams," *Transp. Res. B, Methodol.*, vol. 40, no. 3, pp. 251–264, Mar. 2006.

[2] J. A. Laval, "Linking synchronized flow and kinematic waves," in *Traffic and Granular Flow*. Berlin, Germany: Springer, 2007, pp. 521–526.

[3] A. Ghazy and T. Ozkul, "Design and simulation of an artificially intelligent VANET for solving traffic congestion," in *Proc. 6th Int. Symp. Mechatronics Appl.*, Mar. 2009, pp. 1–6.

[4] S. Bundesamt, "Destatis," Statistisches Bundesamt, Wiesbaden, Germany, Tech. Rep. 5462403177004, 2018.

[5] K. Griesbach, "Lane change prediction in the urban area," Ph.D. dissertation, Dept. Theor. Phys., Especially Comput. Phys., TU Chemnitz, Chemnitz, Germany, 2019.

[6] V. Leonhardt and G. Wanielik, "Recognition of lane change intentions fusing features of driving situation, driver behavior, and vehicle movement by means of neural networks," in *Advanced Microsystems for Automotive Applications 2017*. Cham, Switzerland: Springer, 2018, pp. 59–69.

[7] V. Leonhardt and G. Wanielik, "Neural network for lane change prediction assessing driving situation, driver behavior and vehicle movement," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 1–6.

[8] F. Bocklisch, S. F. Bocklisch, M. Beggiato, and J. F. Krems, "Adaptive fuzzy pattern classification for the online detection of driver lane change intention," *Neurocomputing*, vol. 262, pp. 148–158, Nov. 2017.

[9] W. Yuan, Z. Li, and C. Wang, "Lane-change prediction method for adaptive cruise control system with hidden Markov model," *Traffic Saf. Intell. Transp. Syst.*, vol. 10, no. 9, pp. 1–9, 2018.

[10] J. Li, B. Dai, X. Li, X. Xu, and D. Liu, "A dynamic Bayesian network for vehicle maneuver prediction in highway driving scenarios: Framework and verification," *Electronics*, vol. 8, no. 1, p. 40, Jan. 2019.

[11] I-FAS, "Carai-studie [raw data files]," Technische Universität Chemnitz, Chemnitz, Germany, Tech. Rep., 2013.

[12] M. Beggiato and J. F. Krems, "Sequence analysis of glance patterns to predict lane changes on urban arterial roads," in *Proc. 6th Tagung Fahrerassistenzsysteme*, 2013.

[13] A. Goudarzi and C. Teuscher, "Reservoir computing: Quo vadis?" in *Proc. 3rd ACM Int. Conf. Nanosc. Comput. Commun.*, Sep. 2016, pp. 1–6.

[14] M. D. Skowronski and J. G. Harris, "Automatic speech recognition using a predictive echo state network classifier," *Neural Netw.*, vol. 20, no. 3, pp. 414–423, Apr. 2007.

[15] H. Jaeger, "The 'echo state' approach to analysing and training recurrent neural networks-with an erratum note," German Nat. Res. Center Inf. Technol., Bonn, Germany, Tech. Rep., 2001.

[16] M. Lukoševičius, *A Practical Guide to Applying Echo State Networks*. Berlin, Germany: Springer, 2012.

[17] S. Su, K. Muelling, J. Dolan, P. Palanisamy, and P. Mudalige, "Learning vehicle surrounding-aware lane-changing behavior from observed trajectories," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1412–1417.

[18] C. Olah. (2015). *Understanding LSTM Networks*. Accessed: Jun. 12, 2019. [Online]. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[19] M. Abadi *et al.* (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. [Online]. Available: https://www.tensorflow.org/

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[21] T. Wakasugi, "A study on warning timing for lane change decision aid systems based on driver's lane change maneuver," in *Proc. Int. Tech. Conf. Enhanced Saf. Vehicles*. Washington, DC, USA: National Highway Traffic Safety Administration, 2005, pp. 1–7.

Karoline Griesbach was born in Dissen am Teutoburger Wald, Germany. She received the B.S. and M.S. degrees in sensor technology and cognitive psychology from TU Chemnitz, Chemnitz, Germany, in 2014 and 2015, respectively, where she is currently pursuing the Ph.D. degree with the professorship of theoretical physics, in particular computational physics. She is currently a Research Assistant. Her research interests include intelligent transportation systems, machine learning, and driving behavior prediction.

Matthias Beggiato was born in Italy. He received the degree in psychology and educational science from the University of Vienna in 2005. He has a background in computer science. Since 2010, he has been a Researcher with the Department of Cognitive and Engineering Psychology, Chemnitz University of Technology, and a Post-Doctoral Researcher, since 2015. He is currently a Team Leader of the Traffic Psychology Group. His current research interests include human–machine interaction issues related to assistance and automation, interaction between vulnerable road users and automated vehicles, comfort in automated driving, and the use of databases for analyzing large datasets in transportation research.

Karl Heinz Hoffmann was born in Moers, Germany. He received the Diploma degree in mathematics, the Diploma degree in physics, and the Dr. rer. nat. degree from the RWTH Aachen in 1978, 1979, and 1982, respectively, and the Habilitation degree in physics from the University of Heidelberg in 1988. Since 1993, he has been a Professor of Physics, in particular Computational Physics, with the Technical University Chemnitz. His research interests include statistical physics and classical thermodynamics, both with a focus on non-equilibrium processes and AI methods.