2021 IEEE Intelligent Vehicles Symposium (IV)
July 11-17, 2021. Nagoya, Japan

Trajectory-Based Failure Prediction for Autonomous Driving

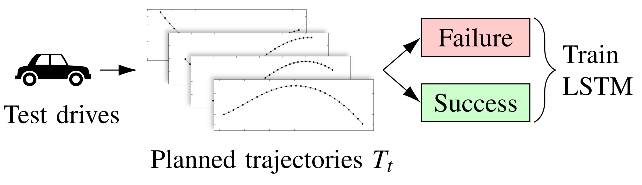
Christopher B. Kuhn[1,2], Markus Hofbauer[1], Goran Petrovic[2], and Eckehard Steinbach[1]

Abstract— **In autonomous driving, complex traffic scenarios can cause situations that require human supervision to resolve safely. Instead of only reacting to such events, it is desirable to predict them early in advance. While predicting the future is challenging, there is a source of information about the future readily available in autonomous driving: the planned trajectory the car intends to drive. In this paper, we propose to analyze the trajectories planned by the vehicle to predict failures early on. We consider sequences of trajectories and use machine learning to detect patterns that indicate impending failures. Since no public data of disengagements of autonomous vehicles is available, we use data provided by development vehicles of the BMW Group. From over six months of test drives, we obtain more than 2600 disengagements of the automated system. We train a Long Short-Term Memory classifier with sequences of planned trajectories that either resulted in successful driving or disengagements. The proposed approach outperforms existing state-of-the-art failure prediction with low-dimensional data by more than 3% in a Receiver Operating Characteristic analysis. Since our approach makes no assumptions on the underlying system, it can be applied to predict failures in other safety-critical areas of robotics as well.**

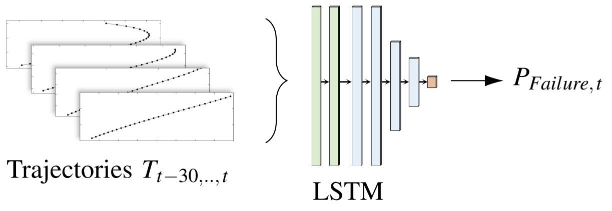
I. INTRODUCTION

Autonomous driving has received significant focus over the last years. However, situations in which human intervention is required to consistently guarantee safety currently cannot be avoided. Such situations can be caused by complex traffic, novel environments or failures of individual components of the system, causing the system to disengage and hand over control to the safety driver. Studies have shown that it takes human drivers around one second to react to a disengagement of the autonomous system [1] and up to 40 seconds to fully stabilize control of the car [2]. It is desirable to anticipate such events as early as possible. This can be approached by learning patterns from previous failures in order to predict future ones. Some existing methods learn to detect deviations in the steering angle [3], [4], while other works monitor the entire car state [5] or sensory input such as camera images [6]. In this work, we propose to learn from a previously disregarded source of information: the planned trajectories generated by the vehicle.

The trajectory the car intends to drive is a readily available source of information about planned future behavior of the vehicle. Monitoring the steering angle or the state of the car only allows to detect difficult situations after the car has already reacted to them. We argue that failure predictions can be made even earlier by analyzing the planned trajectory that ultimately leads to the car state changing. We propose to use machine learning to detect patterns in sequences of planned trajectories that indicate an impending failure. The proposed method consists of two steps. An overview is shown in Figure 1.



(a) Data generation and training



(b) Failure prediction

Fig. 1: Overview of the proposed approach.

In the first step (Fig. 1 (a)), we extract driving sequences from recorded test drives of autonomous vehicles that led to a failure. As failures, we consider disengagements of the autonomous system or the intervention of the human safety driver. We also obtain successful driving sequences to get a balanced data set. From each sequence, we store the planned trajectory of the car at every time point. Then, we train an LSTM (Long Short-Term Memory) classifier with sequences of trajectories from both success and failure sequences.

In the second step (Fig.1 (b)), the trained model is used to classify the current sequence of the last 30 planned trajectories as either being successful or leading to a failure. We evaluate the approach on data collected by development vehicles of the BMW group driving in their autonomous mode. From the provided data, we obtain over 2600 disengagements. The proposed approach outperforms the accuracy of the state-of-the-art failure prediction method from [5] by 2 %. Seven seconds before a failure, the trajectory-based approach outperforms the reference method by almost 6 %, showing the potential of analyzing planned trajectories for predicting far into the future.

The rest of the paper is structured as follows. In Section II, we give an overview of related work. In Section III, we present our trajectory-based failure prediction approach. We discuss the experiments conducted to evaluate the proposed method in Section IV. Section V concludes the paper.

[1]Department of Electrical and Computer Engineering, Technical University of Munich, 80333 Munich, Germany {christopher.kuhn, markus.hofbauer, eckehard.steinbach}@tum.de
[2]BMW Group, Knorrstraße 147, 80788 Munich, Germany {christopher.kuhn, goran.petrovic}@bmw.de

2021 IEEE Intelligent Vehicles Symposium (IV) | 978-1-7281-5394-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/IV48863.2021.9575937

II. RELATED WORK

Failure prediction can be approached in multiple ways. Here, we give a brief overview of the main concepts relevant to this work and which ones we build on.

Uncertainty of neural networks. First, the estimated uncertainty of a model can be used since an uncertain model is more likely to fail. Uncertainty of neural networks has been studied extensively over the last years [7]. Blundell et al. [8] proposed to learn distributions instead of singular values for each weight to allow for a probabilistic model. Hendrycks et al. [9] suggested to calibrate the output of a neural network to correspond more closely to the actual success probability. Besides obtaining the uncertainty from the model itself, some works suggest to add a separate uncertainty estimation module. Corbière et al. [10] proposed to add a confidence estimation network to the end of a given model. In [11], a separate hardness predictor is trained to predict for which input a model will likely make mistakes. Different from those works, we predict failures of an entire self-driving system instead of just a single neural network.

Disagreement-based uncertainty. Another approach is to use the disagreement between multiple models to estimate uncertainty. Monte Carlo dropout [12] generates multiple forward passes with different dropout masks to effectively approximate an ensemble of different models. The variance of that ensemble corresponds to the model's uncertainty. Deep Ensembles [13] follow a similar idea, creating an ensemble by training the same architecture multiple times with different initial weights. The variance of the resulting ensembles is again used to estimate how certain the model is about a new input. Instead of obtaining an ensemble from one single architecture, Ramanagopal et al. [14] detect failures of a perception module by running a second, different perception module in parallel and looking for disagreements between the two modules. A similar approach is followed in [3], where two self-driving systems run in parallel and a high disagreement between them is used to predict an impending failure of the autonomous vehicle. The proposed approach is similar to methods that rely on the disagreement between different predictions. Instead of looking for disagreement within a distribution of models at one time point, we take the same model and look for variations between multiple predictions over time. Our approach can be partially considered as measuring the disagreement of the model with itself over time to predict failures.

Introspection. The idea of learning from previous failures to predict future ones is inspired by the concept of introspection as introduced by Daftry et al. [15]. Introspective failure prediction has since been used for obstacle detection [16], semantic segmentation [17] and disengagement prediction [5], [6]. We follow this fundamental principle in this work.

Future prediction. Besides predicting that a current output of a model is a failure, we are also interested in predicting a failure as early in advance as possible. Hecker et al. [4] predicted mistakes made by an end-to-end steering angle prediction network two seconds in advance, while Michelmore et al. [18] monitored the uncertainty of an end-to-end network to predict crashes in a simulator around four seconds in advance. For our task of predicting disengagements of an actual autonomous system on real roads, only few previous works exist. Fridman et al. [3] used the disagreement between an end-to-end network and the Tesla autopilot to predict disengagements of the Tesla autopilot on highways up to five seconds in advance. Introspective failure prediction based on state data [5] and later also camera data [6] achieved state-of-the-art performance for disengagement prediction up to seven seconds in advance. In this work, we focus on low-dimensional data and therefore compare to the approach from [5], which uses sequences of state data to predict disengagements.

Trajectory generation. Finally, we briefly summarize existing work on trajectory generation in autonomous driving since we propose to use the trajectories planned by the vehicle as our main source of information. Related approaches include imitating human driver behavior [19], using a motion model [20] or relying on deep learning [21]. While some approaches consider failures by including collision prediction [22] or generating uncertainty-aware trajectories [23], our approach does not investigate whether the current trajectory itself is problematic, but instead learns to detect patterns within the sequence of computed trajectories that indicate a failure. To the best of our knowledge, this has not been used as a source for uncertainty estimation or failure prediction before. It is applicable to any system where trajectories for the future path are regularly computed and where failure events such as disengagements occur. While we evaluate our approach in the context of autonomous driving, the application to other fields of robotics is straightforward.

III. TRAJECTORY-BASED FAILURE PREDICTION

In this section, we present the proposed trajectory-based failure prediction method. First, we introduce the proposed concept of learning from trajectories that have led to failures before. Then, we summarize the proposed deep learning architecture. Finally, we discuss the data set used to evaluate the proposed approach.

A. Concept

We use the idea of introspective failure prediction which was originally proposed in [15] and later applied to autonomous driving in [5], [6]. First, failures of a given system are recorded. Any event that deviates from a desired state can be defined as *failure*. In this work, we define a failure to be the disengagement of the autonomous system. This can be caused either by the car itself returning control to the human or by the human safety driver intervening to always ensure safety. Successful behavior of the system is recorded as well to obtain a balanced data set.

Any information about the failure event can be used as training data. In previous works, sequences of state data [5] or images [6] were used. Images require complex architectures to be processed. In this work, we instead focus on low-dimensional data to avoid the significant overhead

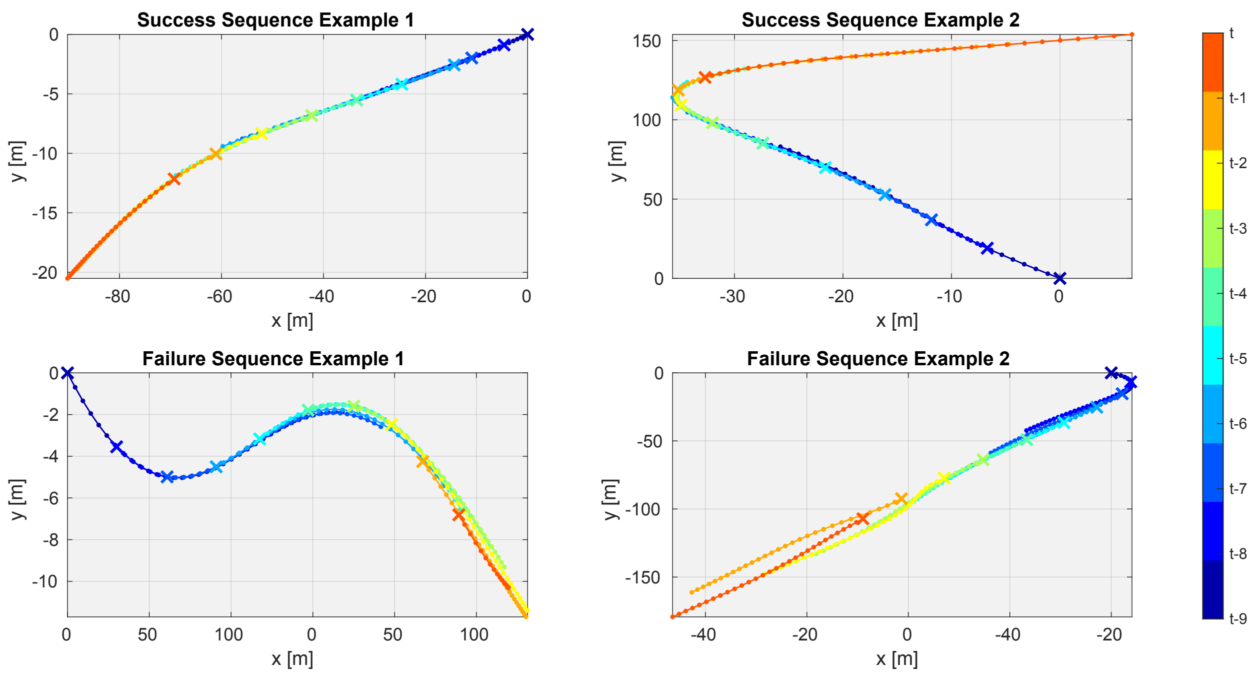


Fig. 2: Visualization of two exemplary failure and success sequences. From each 10 s sequence, every tenth trajectory is plotted in world coordinates. The start of each planned trajectory is denoted by an "X". The colors indicate the time points, dark blue being the beginning and red being the moment of disengagement or the end of the successful sequence. The trajectories from failure sequences show noticeable variations as the disengagement is approached.

of processing images. We propose to use the trajectories computed at each time step by the car as data to learn from. The trajectories we use in this work are generated at a frequency of 10 Hz. Each trajectory contains the next 30 planned time-fixed points of the route. Following [6], we use sequences of 10 seconds before each failure. Successful sequences are evenly sampled at the same length of 10 seconds. As a result, each sequence S_n consists of 100 trajectories T_t^n of the form

$$T_t^n = [p_1, p_2, ..., p_{30}], \quad p \in \mathbb{R}^2, \quad t \in [1, 100]. \qquad (1)$$

After recording N_{fail} failure sequences and sampling N_{success} success sequences, the data set therefore consists of a set of sequences S_n given as

$$S_n = \{T_1^n, T_2^n, ..., T_{100}^n\}, \quad n \in [1, N_{\text{fail}} + N_{\text{success}}]. \qquad (2)$$

All trajectories in a given sequence are labeled as *success* if the sequence was successful driving or labeled as *failure* if the sequence ended in a disengagement.

To visualize the potential of using such trajectory sequences for failure prediction, we show two failure and two success sequences in Figure 2. Ten trajectories at one second time intervals are plotted from each sequence. At the start of each sequence (blue to green), the planned trajectories of all four example sequences largely overlap and do not show visible variations. However, when getting closer to the end of a sequence (yellow to red), the trajectories in the failure sequences start to deviate visibly. This can be explained by a more complex or dynamic traffic situation leading to more drastic changes in the planned route, or in the car not being able to execute the planned trajectory due to new detected obstacles.

It should be noted that the trajectories in successful driving can also vary noticeably when new obstacles are detected and the trajectory is adapted accordingly. However, we argue that the complexity of situations that eventually lead to a disengagement is likely to reflect as more drastic or erratic variations in the planned trajectories. In order to learn which kind of variations are regular and which lead to disengagements, we next propose to train a deep learning based model to distinguish between success and failure cases.

B. Failure Prediction Model

In order to learn from temporal sequences of planned trajectories, we use a recurrent neural network based on Long Short-Term Memory (LSTM) layers. We follow the architecture proposed in [5] which used a combination of LSTM and fully connected (FC) layers to classify sequences of state data as *failure* or *success*. Whereas the state of the car consists of only a few scalar values, our trajectories consist of 30 two-dimensional points. We therefore use a larger and deeper architecture than [5] and propose the model shown in Figure 3. We achieved best results with an architecture consisting of two LSTM layers with 100 nodes each and four FC layers containing 100, 100, 50 and 2 nodes, respectively. A final softmax layer assigns the input a probability of

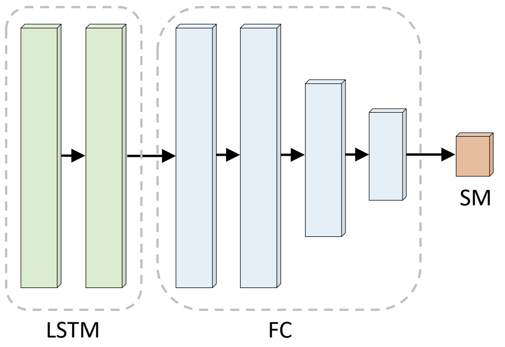


Fig. 3: The architecture proposed for trajectory-based failure prediction. Following [5], the model consists of an LSTM block, a block of fully connected (FC) layers and a softmax (SM) layer.

belonging to a failure or success sequence.

We train this architecture with sequences of 30 planned trajectories. At a frequency of 10 Hz, this corresponds to three second samples, which we evenly sample from the ten second sequences outlined above. From a ten second sequence consisting of 100 generated trajectories, we obtain 71 overlapping samples this way, each consisting of 30 consecutive planned trajectories. Observing time periods of three seconds led to the best performance in previous works [5], [6]. We also apply a moving average filter with a time horizon of 30 predictions to the output of the model. This ensures that a high failure probability is only output if the model consistently predicts failure.

The proposed architecture can be trained with trajectories obtained from any kind of autonomous system as long as a sequence of planned trajectories leading up to a failure event is available. For this work, we evaluate the proposed approach in the context of autonomous driving. Since no public disengagement data is available, we use data provided by the BMW Group. Next, we give an overview of the used data set.

C. Disengagement Data Set

We use data from test drives recorded by fully automated development vehicles by the BMW Group on public roads. From more than six months of test drives, we obtain more than 2600 disengagements. For each disengagement, we use the ten seconds of recorded planned trajectories leading up to the disengagement. Each trajectory consists of the next 30 points and is updated at a frequency of 10 Hz. The ten second sequences are then processed into three second samples as outlined above, yielding over 370 000 samples. We normalize the orientation of each sample by setting the direction of the car at the end of the sample as the x axis and rotating all trajectories accordingly. Each sample is labeled as *failure* if it is from a disengagement sequence and as *success* if it is from a sequence of undisrupted driving. We split the data into 80 % for training, 10 % for validation and 10 % for testing. An overview of the data set is given in Table I.

N_{fail}	2624
Sampled N_{success}	2624
Sequence length	10 s
Planning frequency	10 Hz
Trajectory content	Next 30 points

TABLE I: Summary of the data obtained from test drives performed by BMW development vehicles on public roads.

IV. EXPERIMENTS AND RESULTS

In this section, we discuss the experiments conducted to evaluate the proposed approach and present the results.

A. Data Analysis

First, we analyze the data set by visualizing key characteristics of the trajectories. Since our hypothesis is that trajectories leading to failure show stronger variations, we start by investigating the average curvature of the planned trajectories. We approach this by calculating the angles between the lines spanned by subsequent points of each trajectory. For a trajectory consisting of 30 points p_i, we obtain 28 angles. We sum up the absolute values of the angles to obtain the overall degree of curvature, referred to as curvature for brevity. This process is visualized in Figure 4.

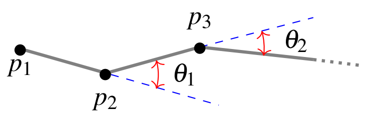


Fig. 4: Calculation of the curvature as the sum of angles $\theta_1, \theta_2, ..., \theta_{28}$ between 30 points $p_1, p_2, ..., p_{30}$ of a planned trajectory.

The average curvature in degree over time for both failure and success sequences is shown in Figure 5. While the curvature is only a simple approximation of the structure of each trajectory, it can be seen that the curvature averaged over all sequences is constant for success sequences while noticeably increasing when a disengagement is approached.

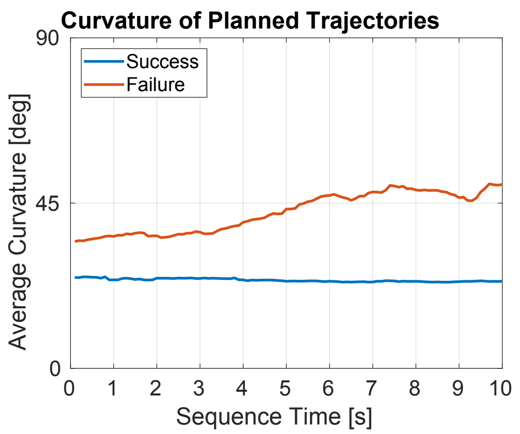


Fig. 5: Visualization of the average curvature of the planned trajectories over time for success and failure sequences.

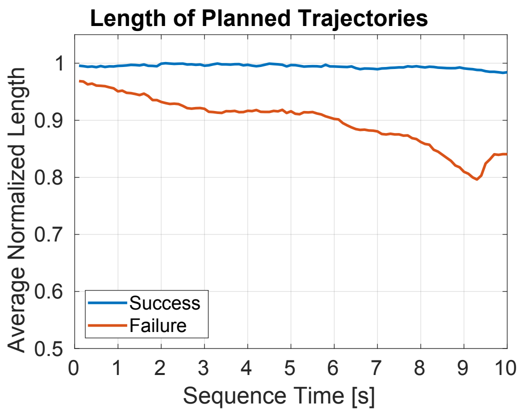


Fig. 6: Visualization of the average normalized length of the planned trajectories over time for success and failure sequences.

Next, we visualize the normalized average length of the planned trajectories over time for both success and failure sequences in Figure 6. Similar to the curvature, the length averaged over all success sequences is mostly constant. For failure sequences, the length starts very close to the average length of success sequences, but declines to around 80 % of the average success length at the point of disengagement. These findings demonstrate the potential of using trajectories to predict that a new driving scene will lead to a failure.

B. Failure Prediction Performance

Next, we present the results of using the proposed method for predicting failures based on planned trajectories. We first compare different methods of extracting information out of trajectories and then compare them to a reference state-of-the-art failure prediction method.

1) Hand-Crafted vs Learned Features: First, we investigate whether the proposed LSTM-based deep network that is trained directly with trajectories is a suitable choice by comparing it to two simpler baseline models that use hand-crafted features. Since the visualizations in the previous paragraph showed that the average curvature and length are visibly different in failure sequences, we first train a Support Vector Machine (SVM) classifier with the average curvature and length of only the current time point as training data. We denote this model as *Curve+Length SVM*. In order to investigate whether considering temporal information is necessary, we next train the LSTM-based architecture from Section III with sequences of the past 30 curvature and length values for each time point, denoted as *Curve+Length LSTM*. We then compare these approaches to our proposed model, referred to as *Trajectory LSTM*, for which we do not manually extract features first, but directly train the proposed neural network with sequences of normalized trajectories. We choose the Receiver Operating Characteristic (ROC) curve as a metric to evaluate the prediction performance on the test set. The results are shown in Figure 7.

The SVM classifier that only uses the curvature and length of the current trajectory as input performs worst at an Area Under Curve (AUC) of 0.71. Learning from sequences of hand-crafted features improves the performance to an AUC of 0.77. The proposed deep network that learns from sequences of entire trajectories performs best by a significant margin, at an AUC of 0.87. This demonstrates that while curvature and length are correlated to failure, they are not sufficient as features for accurate failure prediction.

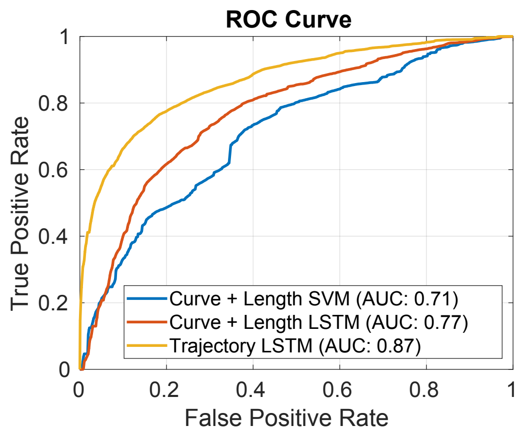


Fig. 7: Comparison of the different evaluated trajectory processing approaches.

2) State vs Trajectory Data: Next, we compare the proposed method to the reference approach of using state data sequences to predict failures from [6], which significantly outperformed previous failure prediction methods such as [3] or [18]. We refer to the reference approach as *State LSTM*. We also combine the two approaches by averaging the predicted failure probability output by the two models, which we refer to as *State + Trajectory*. The ROC curve comparison is shown in Figure 8.

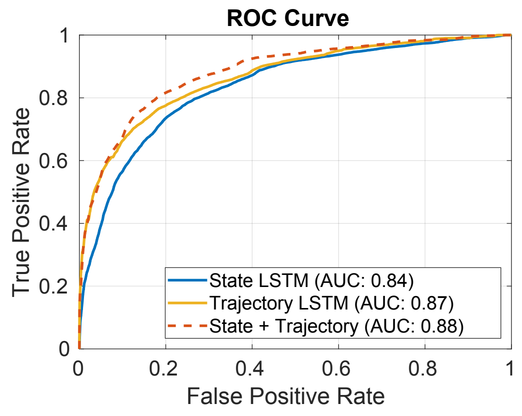
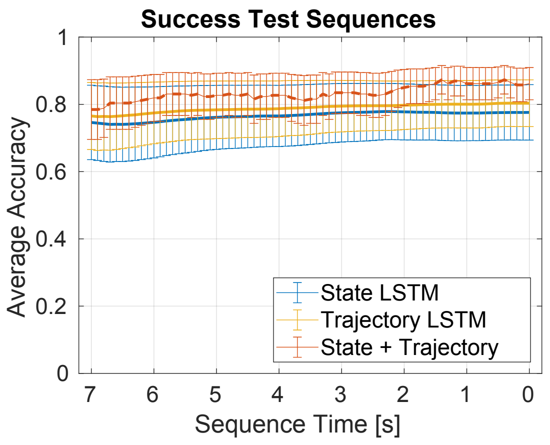
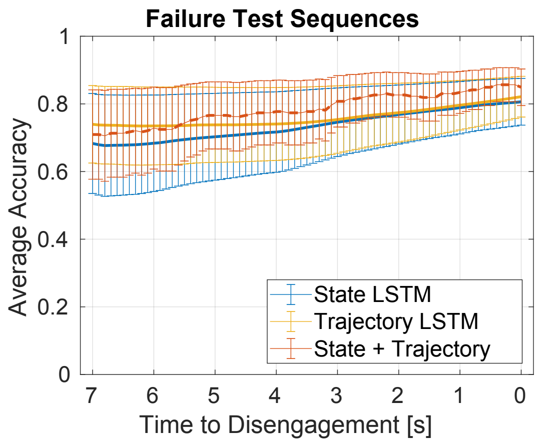


Fig. 8: Comparison of the proposed trajectory-based LSTM, the reference state-data-based LSTM [5] and the combination of both approaches.

The proposed approach achieves an AUC of 0.87 compared to 0.84 for the reference approach, outperforming it by more than 3 %. Interestingly, combining the two methods by



(a) Average prediction accuracy for success sequences.



(b) Average prediction accuracy for failure sequences.

Fig. 9: Average prediction accuracy over time of the compared failure prediction methods for failure sequences (a) and success sequences (b).

simply averaging the scores further improves performance at an AUC of 0.88. This suggests that the two sources of information, state and trajectory data, offer complementary information about the future failures of the car.

Besides the overall performance of the proposed approach, we are also interested in the performance over time. In Figure 9, we show the average accuracy of the proposed method, the state-based approach and the combination of the two over time for both success and failure sequences. The average accuracy over time in success sequences in (a) is mostly constant since no disrupting events occur. On average, the trajectory-based method is 2 % more accurate than the state-based approach. Combining both predictions further improves the performance by another 2 %.

In (b), the performance in failure sequences is shown. The average accuracy increases steadily towards the moment of failure for all approaches. Notably, the trajectory-based approach outperforms the state-based approach by almost 6 % seven seconds before the failure. In the three seconds before the failure, the two approaches converge and achieve the same accuracy. This highlights the key advantage of using trajectory data: Planned trajectories show the planned future behavior of the car and therefore allow to predict failures before the car state is even changed. Only three seconds before the failure does the car state finally reflect what has been detected in the trajectories much earlier.

Finally, we consider the computational complexity of the proposed approach. Since sequences of planned trajectories contain much more data points than state sequences, we used a significantly larger architecture than [5] as outlined in Section III. We compare the number of parameters and the average processing time per prediction in Table II. The proposed trajectory-based approach has eight times as many parameters as the state-based reference approach. The processing time increases by more than 50 % from 3.8 ms to 5.9 ms per prediction. While this is a significant increase, an inference time of around 6 ms is still sufficiently low for real-time applications. This demonstrates the usefulness of using low-dimensional data as compared to images. Even when combining the two methods to achieve the best performance as shown above, the processing time stays below 10 ms per prediction.

Model	**Parameters**	**Processing Time**
State LSTM	2×10^4	3.8 ms
Trajectory LSTM	16×10^4	5.9 ms

TABLE II: Comparison of the number of parameters and average processing time per prediction.

V. CONCLUSION

In this paper, we investigated the planned trajectories of an autonomous vehicle as a low-dimensional source of information about impending failures. We argued that common causes of failure such as complex or dynamic traffic scenarios lead to noticeable variations in the planned trajectories even before the state of the car is changed. We investigated how to best use the information in sequences of planned trajectories to predict failures early on. We have shown that the proposed LSTM-based approach outperforms using hand-crafted features. In an ROC analysis, the proposed method outperformed a state-of-the-art failure prediction approach that also uses low dimensional data by more than 3 %. Seven seconds in advance of a failure, the average prediction accuracy of the proposed approach is almost 6 % higher than the reference approach. This suggests that planned trajectories offer a unique and highly predictive source of information about impending failures.

A limitation of the proposed approach is the increased complexity compared to using state data. Additionally, the introspective idea of learning from previous failures means a significant amount of previous failures needs to be recorded first. However, such data is generated on its own in test drives that need to be performed anyway during development. After training with an initial set of failures, the proposed method can also be continuously fine-tuned and improved once new failure cases are encountered in further test drives.

For future work, the proposed approach could be combined with internal confidence measures of the system to further improve prediction accuracy. Another interesting direction is to investigate how failure predictions based on trajectories could be used to adjust the planned trajectories in advance. This is especially relevant when no human safety driver is present anymore. Instead of alerting a safety driver as done currently, the proposed system can then be used to trigger a correction of the vehicle's strategy to ensure a safe handling of the situation without requiring human intervention.

REFERENCES

[1] V. V. Dixit, S. Chand, and D. J. Nair, "Autonomous Vehicles: Disengagements, Accidents and Reaction Times," *PLOS ONE*, vol. 11, no. 12, p. e0168054, Dec. 2016. [Online]. Available: https://dx.plos.org/10.1371/journal.pone.0168054

[2] N. Merat, A. H. Jamson, F. C. H. Lai, M. Daly, and O. M. J. Carsten, "Transition to manual: Driver behaviour when resuming control from a highly automated vehicle," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 27, pp. 274–282, Nov. 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1369847814001284

[3] L. Fridman, L. Ding, B. Jenik, and B. Reimer, "Arguing Machines: Human Supervision of Black Box AI Systems That Make Life-Critical Decisions," *arXiv:1710.04459 [cs]*, Oct. 2017, arXiv: 1710.04459. [Online]. Available: http://arxiv.org/abs/1710.04459

[4] S. Hecker, D. Dai, and L. Van Gool, "Failure Prediction for Autonomous Driving," *arXiv:1805.01811 [cs]*, May 2018, arXiv: 1805.01811. [Online]. Available: http://arxiv.org/abs/1805.01811

[5] C. B. Kuhn, M. Hofbauer, G. Petrovic, and E. Steinbach, "Introspective black box failure prediction for autonomous driving," in *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2020, pp. 1907–1913.

[6] ——, "Introspective failure prediction for autonomous driving using late fusion of state and camera information," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[7] Z. Ghahramani, "Probabilistic machine learning and artificial intelligence," *Nature*, vol. 521, no. 7553, pp. 452–459, May 2015. [Online]. Available: http://www.nature.com/articles/nature14541

[8] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight Uncertainty in Neural Networks," *arXiv:1505.05424 [cs, stat]*, May 2015, arXiv: 1505.05424. [Online]. Available: http://arxiv.org/abs/1505.05424

[9] D. Hendrycks and K. Gimpel, "A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks," *arXiv:1610.02136 [cs]*, Oct. 2016, arXiv: 1610.02136. [Online]. Available: http://arxiv.org/abs/1610.02136

[10] C. Corbière, N. Thome, A. Bar-Hen, M. Cord, and P. Pérez, "Addressing failure prediction by learning model confidence," *arXiv preprint arXiv:1910.04851*, 2019.

[11] P. Wang and N. Vasconcelos, "Towards Realistic Predictors," p. 16.

[12] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*, 2016, pp. 1050–1059.

[13] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," *Advances in neural information processing systems*, vol. 30, pp. 6402–6413, 2017.

[14] M. S. Ramanagopal, C. Anderson, R. Vasudevan, and M. Johnson-Roberson, "Failing to Learn: Autonomously Identifying Perception Failures for Self-driving Cars," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3860–3867, Oct. 2018, arXiv: 1707.00051. [Online]. Available: http://arxiv.org/abs/1707.00051

[15] S. Daftry, S. Zeng, J. A. Bagnell, and M. Hebert, "Introspective perception: Learning to predict failures in vision systems," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 1743–1750.

[16] S. Rabiee and J. Biswas, "iVOA: Introspective Vision for Obstacle Avoidance," *arXiv:1903.01028 [cs]*, Mar. 2019, arXiv: 1903.01028. [Online]. Available: http://arxiv.org/abs/1903.01028

[17] C. B. Kuhn, M. Hofbauer, S. Lee, G. Petrovic, and E. Steinbach, "Introspective failure prediction for semantic image segmentation," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2020, pp. 1–6.

[18] R. Michelmore, M. Kwiatkowska, and Y. Gal, "Evaluating Uncertainty Quantification in End-to-End Autonomous Driving Control," *arXiv:1811.06817 [cs, stat]*, Nov. 2018, arXiv: 1811.06817. [Online]. Available: http://arxiv.org/abs/1811.06817

[19] P. Liu, A. Kurt, and Ü. Özgüner, "Trajectory prediction of a lane changing vehicle based on driver behavior estimation and classification," in *17th international IEEE conference on intelligent transportation systems (ITSC)*. IEEE, 2014, pp. 942–947.

[20] A. Houenou, P. Bonnifait, V. Cherfaoui, and W. Yao, "Vehicle trajectory prediction based on motion model and maneuver recognition," in *2013 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2013, pp. 4363–4369.

[21] F. Altché and A. de La Fortelle, "An lstm network for highway trajectory prediction," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017, pp. 353–359.

[22] Y. Qian, J. Yuan, and W. Wan, "Improved trajectory planning method for space robot-system with collision prediction," *Journal of Intelligent & Robotic Systems*, vol. 99, no. 2, pp. 289–302, 2020.

[23] X. Huang, S. McGill, B. C. Williams, L. Fletcher, and G. Rosman, "Uncertainty-Aware Driver Trajectory Prediction at Urban Intersections," *arXiv:1901.05105 [cs]*, Jan. 2019, arXiv: 1901.05105. [Online]. Available: http://arxiv.org/abs/1901.05105