ITS4SDC at the ICST 2025 Tool Competition – Self-Driving Car Testing Track

Ali Güllü
Institute of Computer Science
University of Tartu
Tartu, Estonia
ali.ihsan.gullu@ut.ee

Faiz Ali Shah
Institute of Computer Science
University of Tartu
Tartu, Estonia
faiz.ali.shah@ut.ee

Dietmer Pfahl
Institute of Computer Science
University of Tartu
Tartu, Estonia
dietmar.pfahl@ut.ee

2025 IEEE Conference on Software Testing, Verification and Validation (ICST) | 979-8-3315-0814-2/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICST62969.2025.10988983

***Abstract*—Testing and verification of self-driving cars are essential for ensuring their safety and reliability. In the context of the ICST 2024 self-driving cars testing tool competition, we present ITS4SDC, our tool for selecting roads that challenge lane-keeping assist systems by leading the car off the road. ITS4SDC leverages a long short-term memory-based model for the classification of roads as safe and unsafe and subsequently selects unsafe roads for testing.**

***Index Terms*—self-driving cars, lane-keeping assist systems, simulation-based test selection, long short-term memory.**

I. Introduction

Testing self-driving cars (SDCs) is costly; thus, selecting failure-inducing tests from a test suite is important for SDCs. Within the scope of the ICST 2025 SDC testing tool competition, the focus is on selecting roads that cause lane-keeping assistance system failures in SDCs [1].

In this paper, we describe the tool we submitted to the ICST 2025 SDCs testing tool competition, named ITS4SDC (**I**ntelligent **T**est **S**elector **for S**elf-**D**riving **C**ars) [5]. The novelty of our tool lies in treating the entire road as a segment array and using a long short-term memory (LSTM) model to classify roads as safe and unsafe.

II. ITS4SDC Tool Description

A road for the SDCs is represented as an array of 2D (x,y) points. After interpolating the road points, a vehicle equipped with an AI driver in the BeamNG.tech [1] simulator follows the road. The AI driver evaluates the entire road while following its path, adjusting the speed of the SDCs based on the sharpness of bends, making maneuvers, and guiding the vehicle to the end of the road if the maneuvers are successful.

We performed a segment-based analysis of the road, where each segment is defined as a line between two consecutive road points (x,y). For *N* (x,y) road points, a total of *N-1* segments can be defined. Several geometric properties can be analyzed for a given road segment array, such as total segment angles, segment median angles, number of turns, total degree covered, and standard deviation of the segments. However, segment properties alone are insufficient to assess road safety; the position of the segment within the road is also important. A road can have sharp turns, but if the sharp turn occurs at the beginning of the road, the SDCs do not significantly increase speed due to the sharpness level and could successfully track the road without leaving the lane. We considered two segment-based analyses:

- Difference of Segment Angles: The angular displacement formed between a segment defined by two consecutive road positions and an adjacent segment.
- Segment Lengths: The length of a segment is defined by a Euclidean distance between two consecutive road positions.

The importance of consecutive segment relationships led us to adopt the LSTM model. LSTM models are well-suited for sequence classification tasks. Binary labels are used: 0 for "FAIL" and 1 for "PASS".

For a $(2, N)$ road coordinate array, such as $\{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$, the feature extraction yields $(\textit{segment}_{\text{angle}}, \textit{segment}_{\text{length}})$. N points define the road coordinates, starting from the initial point. As a result, a feature vector of size $(2, N-1)$ is produced. This feature vector captures sequential road characteristics, such as changes in angle and segment length, and serves as input for the LSTM layer.

Segment angles are calculated with respect to the two-dimensional Cartesian coordinate system. However, the road's initial rotation varies, resulting in high initial angle values (e.g., 200°or 300°). The absolute angle values are irrelevant as long as the road geometry remains unchanged; only the angular changes between segments are meaningful. Figure 1 illustrates the angles and lengths of the first three road segments relative to the x-axis.

To prevent excessively large numerical values in the model's input vector and to speed up training, the angle value of the first segment is initialized to zero. For subsequent segments, the angle difference is calculated by subtracting the angle value of the preceding segment. This difference is then used as the angle information for the current segment. This approach adjusts the angle values for all segments relative to their preceding segments.

The LSTM model is implemented as a bidirectional LSTM layer. Bidirectional LSTMs not only consider past information but also incorporate future information by processing

[1]BeamNG.tech

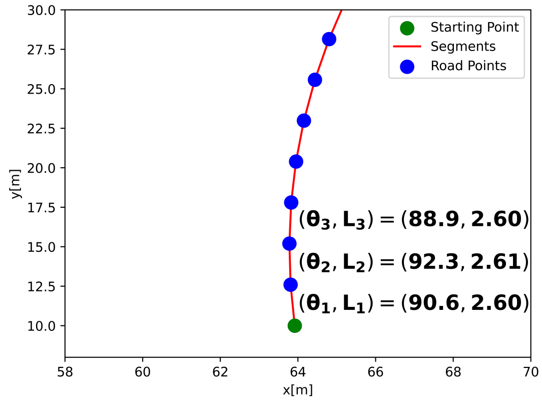

Fig. 1. Angle and length pairs corresponding to the road coordinates

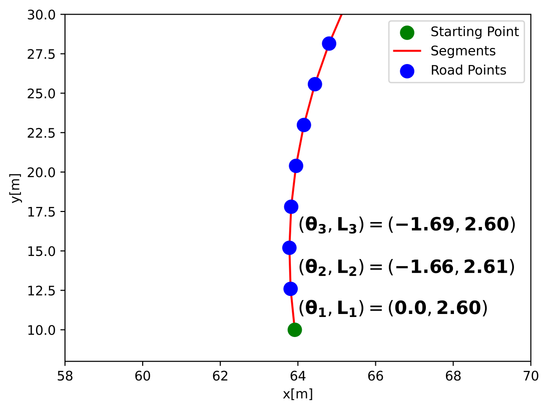

Fig. 2. Adjusted angles and lengths relative to previous reference segments.

sequences in both forward and backward directions. We optimized the LSTM model's architecture through experiments, resulting in a single-layer LSTM model with 220 LSTM cells. The "tanh" activation is applied to LSTM cell states and outputs, followed by a dense layer with a sigmoid activation function. This configuration enabled the prediction of "PASS" (1) and "FAIL" (0) labels as probabilities between 0 and 1.

The flowchart illustrating the working mechanism of the model is presented in Figure 3. The model takes test road coordinates as input and, in the "Segment-Based Feature Extraction" section, computes the input features required for the LSTM layer. It calculates the angle values as the angular difference between each segment and the preceding segment. At the output of the LSTM layer, a dense layer compresses the information from the LSTM layer into a single value. The final sigmoid layer outputs the result as a "probability" between 0 and 1; if the dense layer output exceeds 0.5, the test case is interpreted as "PASS"; otherwise, it is interpreted as "FAIL".

III. Conclusions

In this paper, we introduced ITS4SDC, a test selector tool designed to improve the testing of lane-keeping assist systems in SDCs. ITS4SDC leverages an LSTM model to classify roads as safe or unsafe, using segment angle differences and

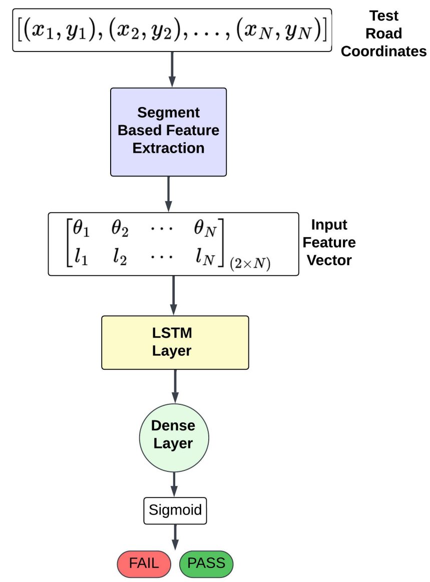

Fig. 3. The flowchart of the ITS4SDC

lengths as feature vectors to identify failure-inducing test cases. The tool effectively detects challenging road segments, contributing to more reliable and cost-efficient testing. Further details on the tool's performance will be available in the competition report.

IV. Acknowledgements

This study was co-funded by the European Union and the Estonian Ministry of Education and Research via project TEM-TA120, by BMK, BMAW, and the State of Upper Austria in the frame of the SCCH competence center INTEGRATE [(FFG grant no. 892418)] part of the FFG COMET Competence Centers for Excellent Technologies Programme, by grant PRG1226 of the Estonian Research Council, and by Bolt Technology ÖU. We gratefully acknowledge the BeamNG company for providing us with the simulation environment that enabled the test case execution and dataset preparation.

References

[1] C. Birchler, S. Klikovits, M. Fazzini and S. Panichella, "ICST Tool Competition 2025 - Self-Driving Car Testing Track", IEEE/ACM International Conference on Software Testing, Verification and Validation (ICST), 2025.

[2] C. Birchler, N. Ganz, S. Khatiri, A. Gambi and S. Panichella, "Cost-effective simulation-based test selection in self-driving cars software with SDC-Scissor", 2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 2022, pp. 164-168.

[3] C. Birchler, S. Khatiri, B. Bosshard, A. Gambi and S. Panichella, "Machine learning-based test selection for simulation-based testing of self-driving cars software", Empirical Software Engineering, vol. 28, no. 3, p. 71, 2023.

[4] C. Birchler, C. Rohrbach, H. Kim, A. Gambi, T. Liu, J. Horneber, T. Kehrer and S. Panichella, "TEASER: Simulation-Based CAN Bus Regression Testing for Self-Driving Cars Software", 2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE), 2023, pp. 2058-2061.

[5] ITS4SDC Tool, Available: https://github.com/christianbirchler-org/sdc-testing-competition/tree/main/tools/ITS4SDC . [Accessed: 03-Jan-2025].