

Attacks on Autonomous Driving Systems: Automated Model-based Detection

EECE 571K: Security and Reliability in IoT - Project Proposal

Aarti Kashyap

Student number: 19115724

Electrical and Computer Engineering, UBC
akashyap18@ece.ubc.ca

Abdul Rehman Anwer

Student number: 50342690

Electrical and Computer Engineering, UBC
abanwer@ece.ubc.ca

Abstract—Autonomous vehicles (AV) have seen an immense increase in popularity recently and are expected to become fully operational on normal roads in near future. The safety critical nature of autonomous driving systems (ADS) of AVs makes quantifying the resilience properties of such systems highly important, for ensuring public and governmental support for their adoption. Resilience of DNN portion of ADS has been greatly studied but effect on reliability of ADS due to other components such as sensors present in ADS still remains to be explored. Prior work in this area focuses on using Fault Injection without any systemic approach to assess the safety and reliability of autonomous vehicles which also test the sensors present in the system do not provide good test coverage due to diverse range of possible inputs for control systems of ADS. We propose using a systematic approach to find vulnerabilities in ADS introduced due to sensors and compare its performance overhead and coverage with fuzz testing techniques.

I. INTRODUCTION

Fully automated self driving vehicles are poised to take the roads in near future. Partial self driving systems have already been deployed and tested for over a billion miles [2], providing valuable sensor data that is used to improve the performance of automated driving systems. The adoption of autonomous driving systems (ADS) on a large scale in real world is highly dependent on their reliability. Moreover, for the acceptance of deployment of such safety critical systems by wider public, makes it necessary to understand the resilience profiles of autonomous driving systems, identify and improve weaknesses in current designs and exhaustively test these fixes to increase public confidence.

Autonomous Driving System are composed of multiple sub-systems, the major parts being the sensor arrays and the control systems. A wide range of sensors like RGB cameras, radars, lidars, sonar, GPS etc. are used to provide information about the environment to the control system of ADS, which usually uses a Deep Neural Network to find the best course of action and gives commands to actuators, to control the movement of the vehicle. Such a highly integrated systems introduce various concerns about the overall reliability of the system. Moreover as ADS are highly safety critical, regulations and standards impose stringent reliability requirements

on their components. For example the SDC FIT rate (Failure-in-Time rate) for of SoC used for DNN inferencing specified by ISO-26262 is 10 i.e. ten failures in one billion hours of operation [5]. These strict reliability requirements, demand that a reliability analysis be performed on integrated systems and vulnerabilities or faults that can propagate, from one sub-system to other sub-systems be identified and mitigation techniques applied to make the system resilient to such faults.

Studying the autonomous driving system is hindered by resource intensive nature of training and testing solutions in real world scenarios. Dosovitskiy et al. [1] recently introduced CARLA, which is an open source simulator for training and validating autonomous driving platforms. It provides an urban environment setting from which data is captured using a flexible array of sensors and signals that are normally present in ADS, this simulated data can then be used to train and test different sub-systems of an ADS. As CARLA also provides sensor reading in addition to images captured by camera, it can be used to study the effect of a vulnerability present in a single component (for example the LIDAR sensor) on overall reliability of the whole ADS. CARLA operates in a server-client model. Server simulates the environment and generates readings for different sensors which are then provided to the client which acts as a driving agent (DA) taking different control decisions based on the provided sensor inputs. Separation of server and client functionality permits implementation of a wide range of DAs.

II. RELATED WORK

Reliability and safety of individual components in ADS have been previously studied. Li et al. [5] studied the impact of soft errors on DNN using Fault injections on hardware accelerators and proposed solutions for making DNNs more resilient. Using Fault injections for testing reliability of system has inherent disadvantage that for complex systems like DNNs a huge number of FI are required to get statistically meaningful insights, moreover the coverage that FI provides is also dependent on the input used for FI, so in case of DNN where the input has a diverse range, performing FI to get meaningful resilience profile of a DNN sub-system of an ADS

can be quite resource intensive, otherwise some corner case may be missed during the analysis. Keeping these things in mind Pi et al. [6] introduced DeepXplore, which tests deep learning systems in a systematic way by finding the inputs for DL systems, that will result in mismatch in output for different DL systems that provide same functionality. In this way corner cases that are not captured by regular testing and training cycle in neural networks are found thus exposing erroneous behavior of the system without a significant overhead of finding corner cases manually. DeepXplore also finds input that maximize the fraction of neurons activated when tested, thus increasing the tests coverage of DNNs. DeepXplore can provide a good defense against vulnerabilities and erroneous behaviors in DNN part of ADS but for overall reliability of the program other components of the system also need to be analyzed.

Jha et al. [3] developed AVFI which is a fault injector that performs resilience assessment of complete autonomous vehicle(AV) systems. AVFI inject faults in without any systematic approach in AV system which consists of CARLA simulator and a driving agent which performs the control actions. AVFI can inject hardware faults (modeling soft errors), data faults (modeling error in sensor values), timing faults and machine learning faults (errors leading to wrong prediction). Jha et al. [4] developed Kayotee which is a FI based tool to inject faults in an ADS (similar to AVFI) and then classifying errors and safety violations impacting the reliability of autonomous vehicles. Kayotee injects faults in both hardware and software components of proprietary Nvidia DriveWorks platform, in a closed looped environment and then error propagation and masking characteristics of the ADS compute stack are evaluated using a predefined ontology model. Kayotee and AVFI provides a holistic view of the system but its fault coverage is also limited, as only those errors can be studied which have been simulated using FI. As whole system is taken into consideration during holistic analysis, trying to increase the test coverage using FI can lead to state space explosion.

III. BACKGROUND

A. Self-Driving Cars

IV. APPROACH

In this project, we intend to use attacks on ADS, to model faults activated in sensors and the communication channels present in an autonomous driving systems. We intend to study the effect of such fault activation (for example, corruption of reported sensor values, corruption of communication channel between sensor and the control system, unexpected activities in hardware controlling the sensor due to faults in control logic etc.) on the output of the entire ADS system. This is an unexplored area and the first step in this research involves formulating of attack models for individual sensors. These attack models will then be used to systematically inject faults in the system and then find vulnerable states in the system.

We intend to compare the results of this systematic FI with to the existing fault injecting techniques and compare the performance and accuracy of these system by answering following questions.

- Does employing systematic approach to FI increase the performance in comparison to fuzz testing?
- Does benefit obtained in terms of performance justify the effort required for developing systematic approaches for FI?

CARLA will be used as a test platform for development and testing of our approach as it provides a convenient way to instrument and modify the readings of different sensors, as well as the behavior of driver agents that controls the autonomous vehicle.

REFERENCES

- [1] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An Open Urban Driving Simulator. In *1st Conference on Robot Learning (CoRL)*, 2018, 2018.
- [2] Tesla Inc. Vehicle safety report (Q4-2018). https://www.tesla.com/en_CA/VehicleSafetyReport.
- [3] Saurabh Jha, Subho S. Banerjee, James Cyriac, Zbigniew T. Kalbarczyk, and Ravishankar K. Iyer. CARLA: An Open Urban Driving Simulator. In *International Conference on Dependable Systems and Networks Workshops*, 2018. IEEE, 2018.
- [4] Saurabh Jha, Timothy Tsai, Siva Hari, Michael Sullivan, Zbigniew Kalbarczyk, Stephen W. Keckler, and Ravishankar K. Iyer. Kayotee: A Fault Injection-based System to Assess the Safety and Reliability of Autonomous Vehicles to Faults and Errors. In *International Workshop on Automotive Reliability and Test (ART)*, 2018. IEEE, 2018.
- [5] Guanpeng Li, Siva Kumar Sastry Hari, Michael Sullivan, Timothy Tsai, Karthik Pattabiraman, Jeol Emer, and Stephen Keckler. Understanding Error Propagation in Deep Learning Neural Network (DNN) Accelerators and Applications. In *International Conference for High Performance Computing, Networking, Storage, and Analysis (SC)*, 2017. ACM, 2017.
- [6] Kexin Pei, Yinzhi Cao, Junfeng Yang, and Suman Jana. DeepXplore: Automated Whitebox Testing of Deep Learning Systems. In *Symposium on Operating Systems Principles (SOSP)*, 2017, pages 1–18. ACM, 2017.