Safety Analysis of Autonomous Vehicle Systems Software

Mike Camara
University of Tartu, Estonia.
mike.gomes.camara@ut.ee

Abstract

Deep Neural Networks (DNNs) have achieved great success in safely driving vehicles autonomously. However, recent studies have exposed their vulnerability against adversarial attacks. In this design science study, I analyse the tools used to evaluate the quality and correctness of autonomous vehicles DNN models. The study starts with the literature review to understand how to train a neural network to autonomously drive a scaled car in a real-world environment and analyse existing metrics to evaluate performance and define the safety of scaled autonomous cars, which will be used to build a baseline. Finally, I experiment and discuss adversarial attacks and defences alternatives, comparing with the baseline to validate the safety of the self-driving agent.

Keywords - adversarial attacks, adversarial machine learning, autonomous vehicles, driverless cars, self driving cars

1. Introduction

Deep Neural Networks (DNNs) have recently accomplished great success in becoming the state-of-the-art solution for tasks in computer vision such as object detection, image classification and segmentation, often beyond human capabilities [1].

Deep learning models, especially Convolutional Neural Networks (CNNs) have been adopted in autonomous driving vehicles [2]. While driverless cars can help reduce the number of fatalities in car accidents caused by human error, such Deep Learning (DL) techniques have proven to be vulnerable to adversarial attacks [3], which in driver-less cars could lead to catastrophic consequences [4]. Such vulnerabilities must be addressed and mitigated before we can see wider adoption of Machine Learning (ML) models to manipulate the steering and throttle predictions of autonomous cars [5].

The goal of this design science study is to demonstrate how adversarial machine learning can be used to improve the deep learning of an autonomous driving car (figure 1).

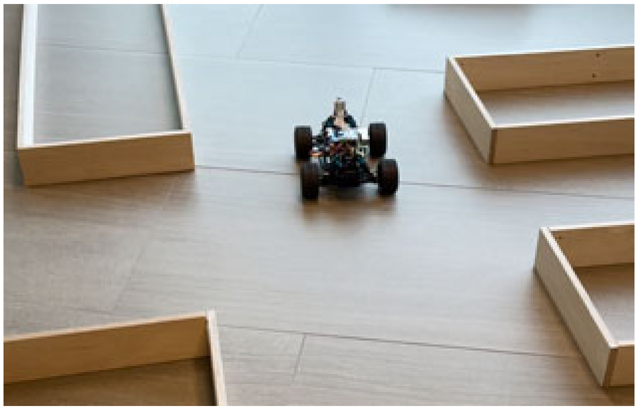

Figure 1. **Donkey Car Platform –** Track for autonomous driving scenario. The ego vehicle travels on the track avoiding collision with the walls and other stationary vehicles set on the track.

The methodology of the design science study will be divided into the five following stages:

- Literature review - how to train a car, analyse overview of existing metrics to evaluate performance. And select an adversarial attack, pick one and check metrics.
- Build baseline for the car.
- Check baseline against adversarial attacks.
- Improve car baseline implementing defences for adversarial attacks.
- Check and compare the performance of improved baseline and check if has created a better car performance.

The report will provide answers to the research questions and demonstrate how using specific adversarial techniques it is possible to make the autonomous vehicle drive safer.

2. Literature Review

Adversarial machine learning techniques is a field of research that involves cyber-security, artificial intelligence and embedded system, therefore this report reviews the publications if the field of software as well as hardware engineering to pick methods to train a car to drive autonomously

Data item	Value	Research Question
Title	Name of the article	
Author v1	List of all author's names	
Year	Publish date	
Machine Learning Model	Tools used to create ML model	**RQ1**
Tests to validate safety	Create baseline for the car	**RQ2**
Adversarial attacks	Adversarial attacks and defences alternatives	**RQ3**
Adversarial defences	Improve baseline	**RQ4**

Table 1. **Data Extraction Table**

and to use adversarial attacks to improve a standard normally trained baseline.

Research Questions.

The research questions aim to find methods for evaluation and implementation of safety in deep learning models for autonomous driving cars.

- **RQ1** – What tools are used to train a DNN to drive a car autonomously in a track?
- **RQ2** – What tools are used to evaluate the quality/correctness of an autonomous vehicle DNN model?
- **RQ3** –What are the mechanisms used to choose among adversarial attacks and defence alternatives to validate the safety of autonomous vehicles?
- **RQ4** – What metrics are used to measure and define the safety of DNNs autonomous vehicles against adversarial threats?

Study selection and quality assessment.

First I defined the keywords that I would include in our search query string.

I aimed to search for relevant papers in the following digital libraries:

- IEEE Xplore
- ACM DL (digital library)
- Scopus
- Springerlink

The selected search string was the following:

(”adversarial attacks” OR ”adversarial machine learning”)
AND (”autonomous vehicles” OR ”autonomous car” OR
”driverless cars” OR ”self driving cars”)

To filter among the candidate papers for the design science study I defined both inclusion criteria such as only consider articles in the field of autonomous driving which were published in the time frame from 2018 to 2021.

I carefully analysed all presented articles and excluded the studies not accessible in full-text, the studies that were duplicates of other studies and studies that did not present tools to validate the safety of DNN models. I only included papers in the English language.

Analysis and Classification.

At this stage of the experiment, I are only considering the results from IEEE Xplore. The search string returned 67 papers in the digital library, but 25 were excluded for being irrelevant to the topic. The remaining 42 articles were read through and the relevant information was extracted into the data extraction table as shown in Table-1.

Training a Neural Network to drive a scaled self driving car.

Chang [6] explores in their study the implementation of multiple variations of deep convolutional neural networks (CNNs) to evaluate the performance of these neural network models in a real-world environment rather than a simulated one. The evaluations of the study are performed in an open source scaled self driving car platform called Donkeycar. The study concludes that a CNN can successfully be trained to recognize visual patterns from a single inexpensive monocular camera as an input and output the commands to pilot the scaled car including steering angle and throttle. The study concludes that a recurrent neural Network (RNN) model was able to navigate the circuit track with the best performance, which was defined by the capacity of the model to avoid deviation from the circuit track and complete determined routes.

Bechtel et al. [7] describes in their study the use of a deep convolutional neural network to perform the commands for the steering angles of a low-cost autonomous car platform which was able to react in real-time to the input of the camera in the real world. The research used a neural network architecture is identical to the one used by NVIDIA's real self driving car DAVE-2. The architecture consists of a CNN model comprised of 9 layers, 250K parameters, and approximately 27 million connections that take a raw color image as input and outputs a single model responsible to prompt the steering angle value for the car.

Mahmoud et al. [8] conducted a case study in they tested different network models available to demonstrate that they could improve the safety of a scaled self driving vehicle while also improving the speed of route completion using techniques to downscale the input image size. The authors explained how neural network tools such as Keras, that runs in Tensorflow and enables the use of high-level neural network API with different architecture structures, can be used in association with the Donkeycar platform to increase the speed of the vehicle on the real-world track circuit reducing the time taken to complete the route.

Implementing adversarial attacks and defences methods.

In [9], Piazzesi et al. explained how trained agent models should be exposed to extensive testing campaigns prior to production deployment as adversarial machine learning attacks and how faults injected can negatively impact the safety of a self driving car. They explained the use of the IBM tool Adversarial Robustness Toolbox (ART) used to launch an experimental adversarial attacks campaign which resulted in safety threats to the scaled self driving car as well as an increase in committed traffic offences.

Alvarez et al. [10] conducted an experiment in which the ART tool was chosen to carry out different attacks to multiple Keras deep learning architectures. They found that even state-of-the-art networks are vulnerable to adversarial attacks.

A case study of using ART tool was described by Lin et al. [11]. In their research they experiment with the tool to generate adversarial image examples and proposed a defence against state-of-the-art adversarial attacks with an iterative adversarial retraining approach, which resulted in the creation of a more robust architecture that was resilient to a number of selected adversarial attacks.

3. Discussion

This section describes the methodology used in the research. Section 3.1 outlines methods to support the creation of deep neural network models to autonomously drive a scaled self driving car. Section 3.2 elaborates on the creation of a baseline and the definition of safety in the context of a scaled self driving car. Section 3.3 distinguish the methods available to improve the safety of the vehicle by measuring the vulnerability of the DNN model to adversarial attacks. Section 3.4 explains how to evaluate and compare the safety of a DNN model in the context of real-world scaled self driving car.

3.1 – Creating a deep neural network end2end model.

Before experimenting with the safety of a scaled self driving car, first, we have to train a deep neural network to pilot the car autonomously in the circuit. Questions arise in regards to the definition of the concept of safety in the environment of scaled cars and the methods available to train the model. Chang [6] described the implementation of a recurrent neural network that was able to keep the car within 6.1 cm deviation from the human driver's path. The research goal was to create an autonomous vehicle that navigates from point A to B without none or minimal human intervention. The authors described how some of the Tesla manufactured cars use deep neural networks for collision avoidance.

To train the neural network model that drives the vehicle autonomously in the circuit we use the scaled self driving car open source platform Donkeycar (figure 2), which combines a remote control car with an embedded Raspberry Pi 4, running various Python packages such as Tornado, Keras, Tensorflow, OpenCV. We used a technique called imitation learning (figure 3) to train the neural network model. In this method a human uses a remote control to output the steering and throttle commands of the car collecting training samples that will be used later to output a model. Chang [6] reported how using the RNN trained model enabled the scaled vehicle to navigate 20 laps of the track (figure 4), whereas linear models failed in five different attempts(figure 5).

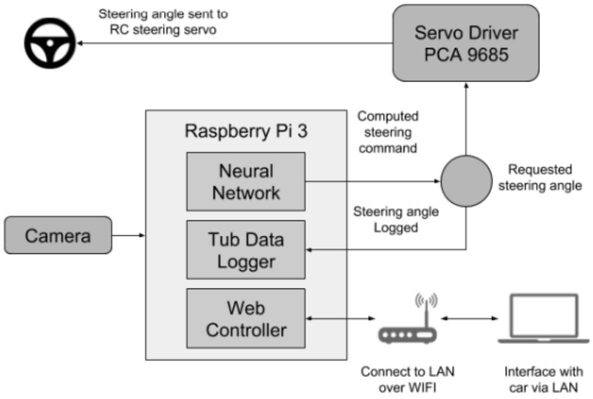

Figure 2. **The Donkeycar platform components –** Figure is cited from [6] .

3.2 – Establishing a baseline and definition of safety.

To validate the performance of a variety of different models Chang [6] measures factors such as the steering accuracy which can be calculated by comparing the deviation from the human commands. Such measurement was achieved using an automated position tracking system implemented by using a ball-tracker in OpenCV. The procedure includes adding a fluorescent pink circle mounted on the car that enables the program to track the global location of the vehicle.

A different approach to the choice of neural network architecture and measurement system was suggested by Mahmoud et al. [8]. In their research, the testing was done exclusively using the Keras Linear and Keras Categorical neural network models. They compared the performance of the model driving the vehicle both in the real world and in a simulation environment. In both cases, the neural networks were exposed to the same amount of training time, which was 10,000 frames and then testing each neural network 5 times for each evaluation criteria, which was lap time and the response rate of the vehicle.

Piazzesi et al. [9] suggested in their paper a number of metrics that could have a direct impact on the safety of an autonomous vehicle. Their work prioritize safety over travelled distance and the success rate is affected by factors such as the number of collisions with the wall and other objects as well as the ignored traffic lights. The target vehicle must reach a destination position B from a starting position A,

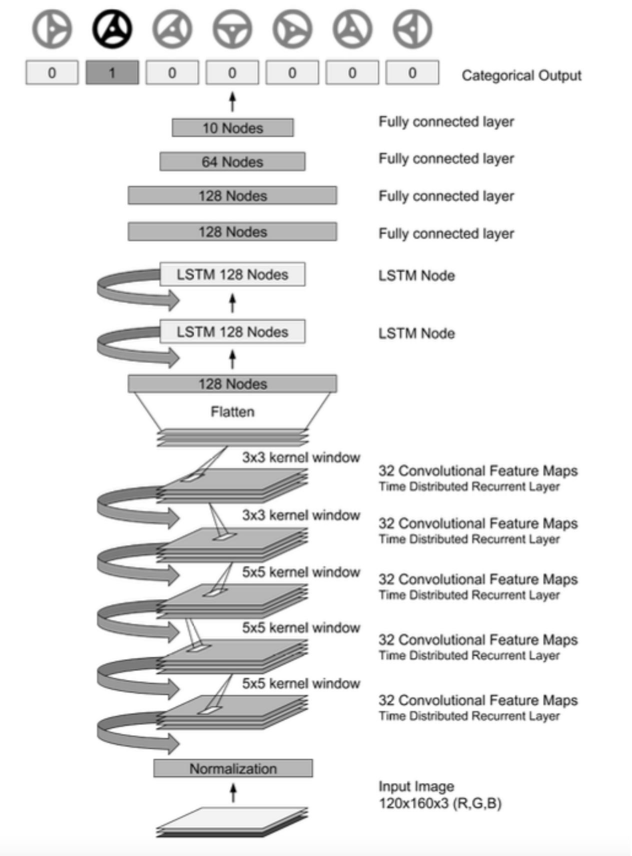

Figure 3. **RNN Categorical output network architecture –** Figure is cited from [6] .

the completion of the task is also counted as a measure of success.

3.3 – Methods available to evaluate the safety of DNN models against adversarial attacks.

Piazzesi et al. [9] explore the ART Toolbox to generate adversarial attacks that led the DNN to make wrong predictions. While adversarial attacks are organized in three categories: evasion, poisoning, and extraction attacks, their study relied only on evasion attacks, because of its likelihood to compromise safety by being carried out on a self-driving agent while it is running. Such attacks can be white-box and black-box attacks, which were both considered in the study. While the white-box attacks require having full access to the architecture and parameters of the model, the black-box attacks do not require having the knowledge on the model structure and architecture. The authors used PytorchFI to inject faults in the trained agents. The study injected four different evasion attacks: Spatial Transformation, HopSkipJump, Basic Iterative Method and NewtonFool. Besides the IBM ART tool, other alternatives to evaluate the robustness of neural networks against adversarial attacks such as CleverHans and Foolbox were mentioned by Assion et al. [12], the authors claimed that adversarial robustness poses a challenge for the deployment of ML-based systems in safety and security-critical environments such as autonomous driving.

3.4 – Evaluation of safety of a DNN model in the context of real-world scaled self driving car.

In the first stage of the experiment, we will execute the DNN model in clean runs without introducing attacks. The results of the clean runs using linear models and RNN models will be described. In our experiment the vehicle could successfully complete the laps five times with a limited number of collisions. Without further applying image augmentation techniques such as cropping the image as suggested in [8], the driving was steady. However, the vehicle still struggling to keep well centered in the lane, crossing the boundaries several times.

The number of collisions as a metric to evaluate the safety of a DNN model is only appropriate for the scaled self-driving agent which uses a single frontal camera and not more sophisticated sensors such as stereo cameras and LiDARs. On the other hand, such attack injection campaigns demonstrated in this study can be repeated with different driving agents [9].

In the second stage, the model is run in the same circuit as the first stage, however, a series of attacks and faults are injected during the run. The results are then compared with the runs of the first stage.

4. Results

To answer RQ-1 I created a track in a B circuit shape and then I trained I convolutional neural network to drive the track. The model proved to be a good driver and performed 14 laps in the circuit without collisions. It also managed to avoid collisions with obstacles placed in the circuit.

To answer RQ-2 I experimented with TensorFlow and Keras which enabled the use of several benchmarks machine learning models including the Google MobileNet used for the stop sign and traffics lights detection. It has also enabled me to use the IBM Adversarial Machine Learning Toolbox, which provided many state-of-the-art machine learning attacks and defences.

To answer RQ-3 the literature review pointed to the use of evasion attacks as it could lead the model to misclassify objects, which can make the autonomous vehicles lose control leading to damage of property or injury. In my experiment I used the Fast Gradient Sign Attack [13].

The final stage of the experiment consists in apply techniques for the defence of the DNN model and then compare the measurements of the final runs against the created baseline defined in stage one (RQ-4). Our experiment focuses on the safety aspects of autonomous vehicle systems, thus,

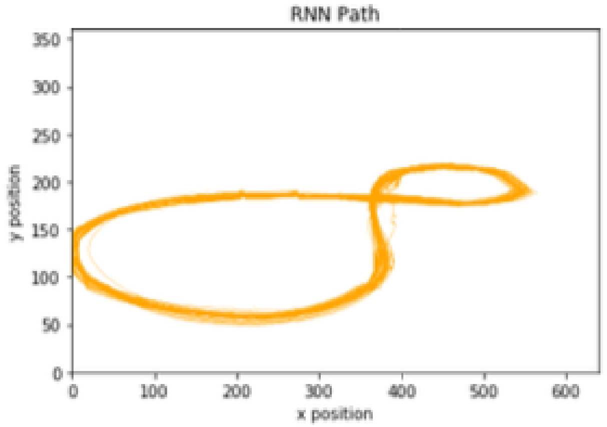

Figure 4. **RNN Path – 8-Track** Figure is cited from [6] .

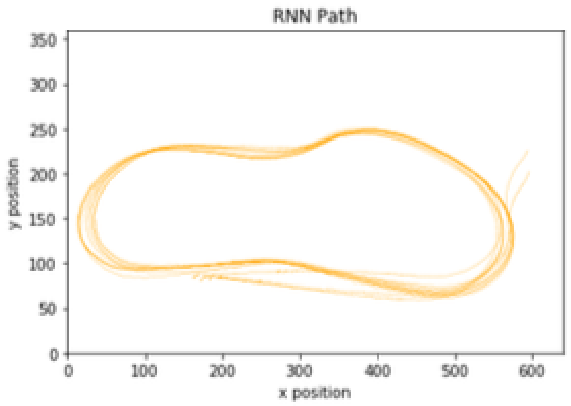

Figure 5. **RNN Path – Circuit** Figure is cited from [6] .

the definition of safety is a crucial aspect of the research and it will be based on the results of the performance of the self driving agent in the circuit track.

5. Conclusions

This report described how to train a neural network to drive a scaled vehicle autonomously in a circuit track. Furthermore, the study discusses methods and metrics that define safety in the context of a scaled self driving agent. Then we demonstrate the injection of attacks in the self driving agent running on the real-world environment using solely open source tools. We expose attacks and faults that impact the performance of the car leading to collisions jeopardizing the safety of the self driving agent. Finally, we implement state-of-the-art adversarial defences that improved the overall performance of the vehicle.

The results are restricted to the environment set in the experiment, nevertheless, they reveal the negative impact of adversarial attacks in neural networks and reinforce the need to test autonomous systems against such threats. The experiments related to this design science report is still in progress. I have so far achieved the goals of creating a model that drives the car autonomously while avoiding objects and detect stop signs and traffic lights. Thus, this report does not yet present clear metrics in regards to the performance and safety of the vehicle. However, I plan to have completed the experiment soon and extend this report to a full design science master's thesis which would be delivered in the upcoming spring.

Finally, the study intended to show methods to test the safety of a self driving agent, using existing open source tools.

References

[1] A. Zolfi, M. Kravchik, Y. Elovici, and A. Shabtai, "The translucent patch: A physical and universal attack on object detectors," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 15232–15241, Computer Vision Foundation / IEEE, 2021. 1

[2] J. Zhang, Y. Lou, J. Wang, K. Wu, K. Lu, and X. Jia, "Evaluating adversarial attacks on driving safety in vision-based autonomous vehicles," *CoRR*, vol. abs/2108.02940, 2021. 1

[3] A. Qayyum, M. Usama, J. Qadir, and A. I. Al-Fuqaha, "Securing connected & autonomous vehicles: Challenges posed by adversarial machine learning and the way forward," *IEEE Commun. Surv. Tutorials*, vol. 22, no. 2, pp. 998–1026, 2020. 1

[4] P. Sharma, D. Austin, and H. Liu, "Attacks on Machine Learning: Adversarial Examples in Connected and Autonomous Vehicles," 2019. 1

[5] S. Pavlitskaya, S. Ünver, and J. M. Zöllner, "Feasibility and suppression of adversarial patch attacks on end-to-end vehicle control," in *23rd IEEE International Conference on Intelligent Transportation Systems, ITSC 2020, Rhodes, Greece, September 20-23, 2020*, pp. 1–8, IEEE, 2020. 1

[6] J. Chang, "Training Neural Networks to Pilot Autonomous Vehicles: Scaled Self-Driving Car," 2018. 2, 3, 4, 5

[7] M. G. Bechtel, E. McEllhiney, M. Kim, and H. Yun, "Deeppicar: A low-cost deep neural network-based autonomous car," in *24th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, RTCSA 2018, Hakodate, Japan, August 28-31, 2018*, pp. 11–21, IEEE Computer Society, 2018. 2

[8] Y. Mahmoud, Y. Okuyama, T. Fukuchi, T. Kosuke, and I. Ando, "Optimizing Deep-Neural-Network-Driven Autonomous Race Car Using Image Scaling," 2020. 2, 3, 4

[9] N. Piazzesi, M. Hong, and A. Ceccarelli, "Attack and fault injection in self-driving agents on the carla simulator - experience report," in *Computer Safety, Reliability, and Security - 40th International Conference, SAFECOMP 2021, York,*

UK, September 8-10, 2021, Proceedings (I. Habli, M. Sujan, and F. Bitsch, eds.), vol. 12852 of *Lecture Notes in Computer Science*, pp. 210–225, Springer, 2021. 2, 3, 4

[10] E. Álvarez, R. Álvarez, and M. Cazorla, "Studying the transferability of non-targeted adversarial attacks," in *International Joint Conference on Neural Networks, IJCNN 2021, Shenzhen, China, July 18-22, 2021*, pp. 1–6, IEEE, 2021. 3

[11] J. Lin, L. L. Njilla, and K. Xiong, "Robust machine learning against adversarial samples at test time," in *2020 IEEE International Conference on Communications, ICC 2020, Dublin, Ireland, June 7-11, 2020*, pp. 1–6, IEEE, 2020. 3

[12] F. Assion, P. Schlicht, F. Greßner, W. Günther, F. Hüger, N. M. Schmidt, and U. Rasheed, "The attack generator: A systematic approach towards constructing adversarial attacks," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 1370–1379, Computer Vision Foundation / IEEE, 2019. 4

[13] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (Y. Bengio and Y. LeCun, eds.), 2015. 4