A Machine Learning Approach for Combating Cyber Attacks in Self-Driving Vehicles

Hunter Berry[1], Mai A. Abdel-Malek[2] and Ahmed S. Ibrahim[2]

[1] Political Sciences and Cyber-Security Law department, Berry College, Georgia, USA
[2] Department of Electrical and Computer Engineering, Florida International University, Miami, Florida, USA
Emails: hunter.berry@vikings.berry.edu, {mabde030, aibrahim}@fiu.edu

***Abstract*— Self-driving vehicles are very susceptible to cyber attacks. This paper aims to utilize a machine learning approach in combating cyber attacks on self-driving vehicles. We focus on detecting incorrect data that are injected into the data bus of vehicles. We will utilize the extreme gradient boosting approach, as a promising example of machine learning, to classify such incorrect information. We will discuss in details the research methodology, which includes acquiring the driving data, pre-processing it, artificially inserting incorrect information, and finally classifying it. Our results show that the considered algorithm achieve accuracy of up to 92% in detecting the abnormal behavior on the car data bus.**

***Index Terms*— Cyber-security, machine learning, self-driving vehicles, gradient boosting.**

I. INTRODUCTION

In recent years, there has been a growing interest in research targeted towards enabling self-driving vehicles [1]–[3]. It is expected that this field will grow by more than 34% a year going into 2023 [4]. While this field's innovations seem to be the pinnacle of future technology, there are numerous obstacles associated with its realization. For example, it has been shown (e.g. in [5]) that self-driving cars are susceptible to hacking, given their sole reliance on data inputs from various sources in their surroundings.

Looking at how self-driving vehicles operate, they use a plethora of information, both from other smart vehicles in the area, and through a number of sensors on the car. The vehicles Controller Area Network (CAN) bus takes in this information and uses it to control the cars movements through the Electronic Control Units (ECUs). However, what does the CAN bus do when it receives information that might be false, and how does it even flag this data to begin with? These various flaws help to raise the question of how these cars can be better secured from both these hackers, and simple mistakes as well. Answering these questions defines the *scope* of this paper.

To answer these questions, we can turn to the idea of *machine learning*. In recent years, machine learning has proved to provide effective solutions to a variety of problems, including cyber-security [6], [7]. For instance, it was shown in [8] that $85-90\%$ accuracy can be achieved while classifying the validity of steering angles within a vehicle. Other studies have also proven effective at locating inaccuracies within data being sent to the CAN bus, including dimensional data (the size of nearby objects) and communication information [9]. Similarly in [10], a device was created to be plugged into a car's CAN bus to intercept various signals and interpret them.

This paper is a prime example of how machine learning can be used in combating cyber attacks in self-driving cars. More precisely, we seek to employ a machine learning algorithm, namely *extreme gradient boosting (XGB)* [11], to show how a machine learning approach can be incorporated into the CAN bus to prevent the use of false data. Furthermore, this paper offers an in-depth discussion of the datasets and models used to train the machine learning algorithm.

Particularly, we will discuss the usage of driving dataset, provided freely by Udacity self-driving car project [12]. We will be utilizing the datsets presenting the steering angles, torque, and speed of the vehicles. Moreover, we will present the process of inserting incorrect data to help in training and testing of the machine learning algorithm. The classification results, presented in this paper, are extremely promising. For instance, we will show that an accuracy of up to 92% is achieved by using a few features that are function of any type of the driving data (e.g. speed).

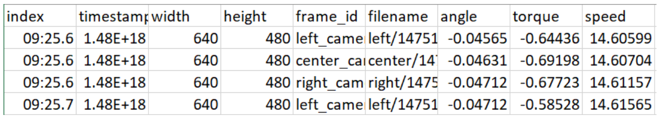
The organization of this paper is as follows. In the next section, we will describe the research methodology consisting of acquiring the driving dataset, modeling cyber attacks, and data pre-processing which is needed for training the machine learning algorithm. Section III presents the machine learning algorithm along with the associated results. Finally, Section IV concludes the paper along with presenting some insights on the future research.

II. Research Methodology

In this section, first we introduce the driving dataset. Second, we will explain the implementation of the cyber attack model. Finally, we discuss the data pre-processing step, which prepares the dataset for the training of the machine learning algorithm.

A. Driving Dataset

In this paper, we have utilized the freely-available driving dataset, published by the Udacity Self-Driving Car team [12]. Udacity's dataset consists of three groups of images, each group represents the driving information capturing a different angle on the car, including left, middle, and right angles. The dataset also includes numerous Comma Separated Value



index	timestamp	width	height	frame_id	filename	angle	torque	speed
09:25.6	1.48E+18	640	480	left_came	left/14751	-0.04565	-0.64436	14.60599
09:25.6	1.48E+18	640	480	center_ca	center/14	-0.04631	-0.69198	14.60704
09:25.6	1.48E+18	640	480	right_cam	right/1475	-0.04712	-0.67723	14.61157
09:25.7	1.48E+18	640	480	left_came	left/14751	-0.04712	-0.58528	14.61565

Fig. 1: Visualization of the Udacity driving dataset [12].

(CSV) sheets, each matching up a specific time to an image frame, as well as other data, including information on the cars torque, speed, steering angle, throttle, brake force, longitude, latitude, and more.

We have used two of the datasets, which were recorded in two different days. The first dataset, recorded on September 29, 2016, consists of twelve minutes and forty seconds of driving data, equating to roughly forty-six thousand rows of data. The second dataset, recorded on October 3, 2016, consists of fifty-eight minutes and fifty-three seconds, equating to roughly two-hundred and twelve thousand rows of data. A brief visualization of the datasets is depicted in Fig. 1.

B. Cyber-Attack Model

Given our goal of being able to locate abnormal data in the car network, we have artificially inserted some incorrect data in some of the rows in the dataset. To simulate an attack where an attacker is attempting to insert an incorrect data into a vehicles CAN bus, we used a simple Python script to insert values into the driving dataset. We used Pythons built-in random library, as well as the Pandas and CSV libraries, to take in a CSV from the dataset and manipulate it. We added a column titled flag to the CSV, representing a Boolean value, where 0 represents a true value, and 1 represents an incorrect value.

Furthermore, we created a new line of artificial data every five to twelve data rows, in which an incorrect value is within the range of $[x-0.8, x-0.4]$ or $[x+0.4, x+0.8]$, where x represents the true value of the previous reading. After inputting the incorrect values into the CVS dataset, it becomes ready be used by a machine learning algorithm.

C. Time-based Data Pre-processing

TABLE I: Features pre-processing.

$f(t)$	$f(t-1)$	$f(t-2)$	$f(t)f(t-1)$	$f(t)f(t-2)$

The dataset is imported into a Pandas data frame, and loaded into variables. At time t, let $f(t)$ denote the feature value which may represent steering angle, speed, or torque. Each feature is then shifted back two times. Thus, instead of a row having simply the value for a steering angle $f(t)$, it will also contain $f(t-1)$ and $f(t-2)$. This allows us to simulate a time-series machine learning model, since our goal is to detect whether or not a data row is injected given the previous values. Finally, difference values of $f(t)-f(t-1)$ and $f(t)-f(t-2)$ are also included within the data row. Table I shows the 5 versions of the current and previous data, which are used in training the machine learning approach.

III. Machine Learning Approach and Results

In this section, we first introduce the machine learning approach considered in this paper. Then, we present the classification results.

A. Machine Learning Approach

In this paper, we have chosen the XGB machine learning approach [13] aiming to identify the injected data rows in the dataset. We have utilized the *XGBClassifier* class, imported from the aforementioned XGBoost [13] library, which is specifically designed for classification problems such as the problem considered in this paper. In this work, we focus on three different types of data, namely, speed, torque, and steering angles.

TABLE II: XGB Parameters.

Parameter	Value
Learning rate	0.1
$n\_estimators$	1000
$early\_stopping\_rounds$	10
max_depth	15

We ran multiple tests to define the optimum values for a set of variables in the XGBClassifier class. Evaluating its accuracy when changing the learning rate on a scale of $[0.1-0.2]$, $n\_estimators$ on a range of $500-1500$, and $early\_stopping\_rounds$ on a range of $5-20$. The best accuracy was established using the values indicated in Table II.

B. Classification Results

All tests were conducted on a Lenovo Yoga 700 11ISK, running the Windows 10 operating system. The system uses an Intel Core m5-6Y54 CPU @ 1.10GHz, 1501 Mhz dual-core processor and a Intel(R) HD 515 graphics card. It had eight gigabytes of Random Access Memory (RAM).

The classification results of XGBClassifier, assuming a single type of data (e.g. steering angle) are shown in Fig. 2. In Fig. 2, x refers to the data at its time, $x-j, j = 1,2,3$ refers to the measured data delayed with j reading interval. Similarly, diff$-i$, $i = 1,2,3$ refers to the difference in measurement between the current reading and the one delayed by i time slots. As shown in Fig. 2, the correct classification of the injected data (i.e. performance accuracy) increases as the number of features increases and it achieves its maximum at 3 features.

For example, depending only on the current reading of one feature (e.g. speed) in the training and testing in the XGBClassifier results in accuracy of 58%. As we add the received data of the same type (e.g. speed) in earlier slots, the accuracy increases to 72%. Finally, a higher accuracy of 92% is achieved by only using the difference in readings of one type of data.

IV. Conclusion and Future Research

Overall, it is evident that machine learning has a lot of potential when it comes to cyber-security and the security

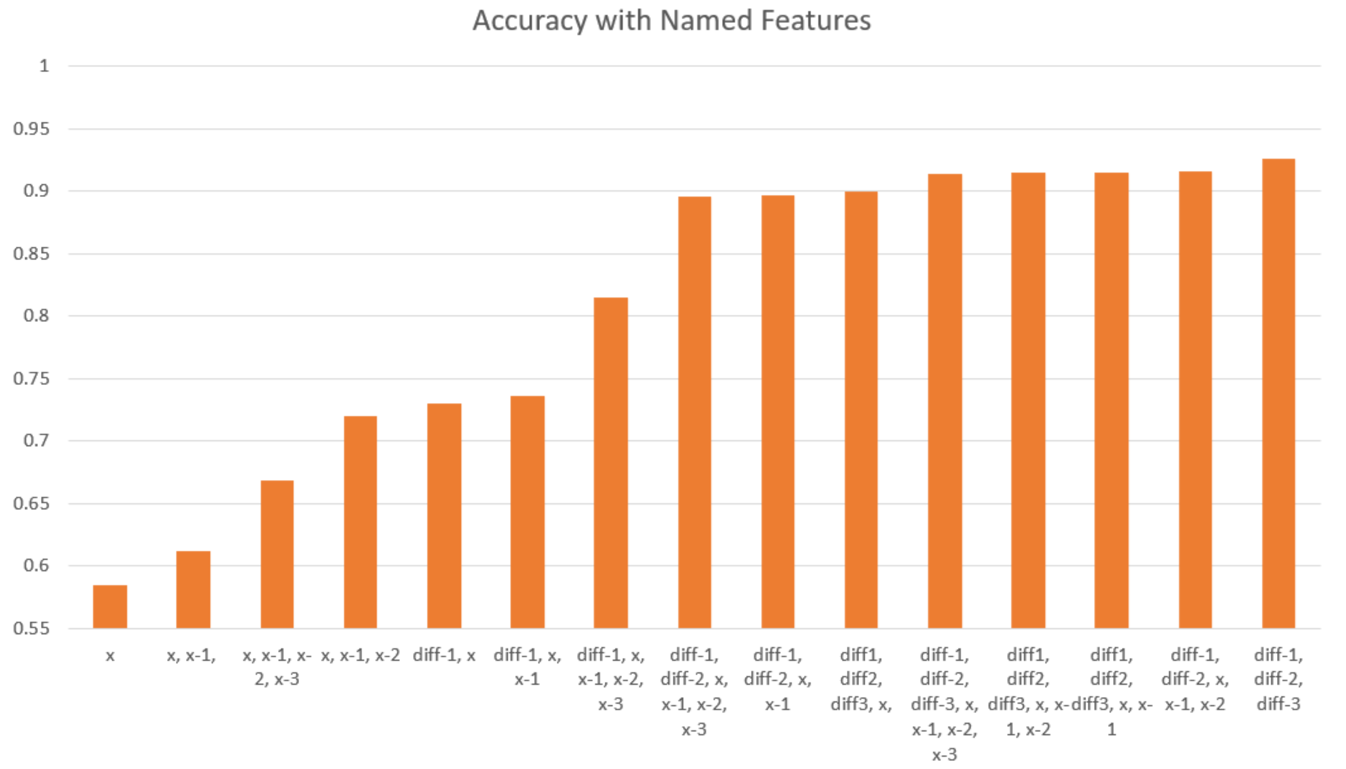


Fig. 2: Detection accuracy dependence on the number of features, extracted from one data type only (e.g. steering angle, speed, or torque).

involved in self-driving vehicles. In this paper, we have shown that the XGB machine algorithm can achieve accuracy of 92% in detecting the abnormal packets on the car CAN bus. Therefore, our algorithm has the potential to be used in a variety of different situations to successfully prevent false information from interfering with the cars ECU.

However, there is always ways to improve these algorithms and increase their accuracy. Although the use of TensorFlows image manipulation and reading libraries were beyond the scope of this study, future research could make use of scalars, combined with machine vision algorithms and various deep learning models, to determine the validity of certain data inputs through image analysis.

References

[1] KPMG, “Self-Driving Cars: The Next Revolution.”

[2] National Science Foundation, “Science of Innovation: Self-Driving Cars.”

[3] E. Guizzo, “How Googles Self-Driving Car Works,” 2011.

[4] Market Watch, “Self-driving Car Market to Grow at a CAGR of 36.2%, leading to global revenue of USD 173.15 Bn by 2023,” 11-Sep-2018, Available https://www.marketwatch.com/press-release/self-driving-car-market-to-grow-at-a-cagr-of-362-leading-to-global-revenue-of-usd-17315-bn-by-2023-2018-09-11, [Accessed: 13-Sep-2019].

[5] A. Greenberg, “Securing Driverless Cars From Hackers Is Hard. Ask the Ex-Uber Guy Who Protects Them,” Wired, 03-Jun-2017, Available https://www.wired.com/2017/04/ubers-former-top-hacker-securing-autonomous-cars-really-hard-problem/, [Accessed: 13-Sep-2019].

[6] R. Das and T. H. Morris, “Machine Learning and Cyber Security,” International Conference on Computer, Electrical and Communication Engineering (ICCECE), Kolkata, 2017, pp. 1-7.

[7] D. C. Le and A. Nur Zincir-Heywood, “Machine learning based Insider Threat Modelling and Detection,” 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), Arlington, VA, USA, 2019.

[8] M. Smolyakov, A. Frolov, V. Volkov, and I. Stelmashchuk, “Self-Driving Car Steering Angle Prediction Based On Deep Neural Network An Example Of CarND Udacity Simulator,” IEEE 12th International Conference on Application of Information and Communication Technologies (AICT), 2018.

[9] J. Monteuuis, J. Petit, J. Zhang, H. Labiod, S. Mafrica and A. Servel, “ “My Autonomous Car is an Elephant”: A Machine Learning based Detector for Implausible Dimension,” Third International Conference on Security of Smart Cities, Industrial Control System and Communications (SSIC), Shanghai, 2018.

[10] A. Greenberg, “Hackers Could Take Control of Your Car. This Device Can Stop Them,” Wired, 03-Jun-2017, Available: https://www.wired.com/2014/07/car-hacker/, [Accessed: 13-Sep-2019].

[11] B. Boehmke and B. Greenwell, “Hands-on Machine Learning with R,” Available https://bradleyboehmke.github.io/HOML/, [Accessed: 13-Sep-2019].

[12] Udacity, “Self Driving Car Driving Dataset,” Available https://github.com/udacity/self-driving-car, [Accessed: 13-Sep-2019].

[13] XGBoost, “XGBoost Documentation,” Available https://xgboost.readthedocs.io/en/latest/, [Accessed: 13-Sep-2019].