Electronic Control Unit (ECU) Identification for Controller Area Networks (CAN) using Machine Learning

Niroop Sugunaraj and Prakash Ranganathan
School of Electrical Engineering and Computer Science (SEECS)
University of North Dakota
Grand Forks, ND-58202

C2SR
CENTER FOR CYBER SECURITY RESEARCH

DECS

Abstract

Electronic control units (ECUs) control several vehicular functionalities such as adjusting seat positions or driver/passenger windows, and wiper and headlight control. Identifying these ECUs is a complex, tedious process and requires extensive reverse engineering of the control area network (CAN) data set, as a modern car may have anywhere from 40 to 150 ECUs. This project uses three machine algorithms (k-nearest neighbors, Gaussian Naive Bayes, and Decision Tree) to classify five different ECU signatures. Cross-validation confirms that Decision Tree and k-nearest neighbors models generalize well to CAN data.

Introduction

In 2011, the average automobile was estimated to have at least 40 ECUs and more than 10 million lines of code. These ECUs were primarily responsible for improving engine performance, fuel consumption, and vehicle safety. As of 2020, there are approximately at least 80-150 ECUs in a standard vehicle responsible for handling various vehicular functions such as lane keep assist, self-parking, cruise control, etc. [1-2]. These ECUs periodically broadcast the data they collect to other ECUs on the vehicle through a network called controller area network (CAN) which is a multi-master broadcasting bus that uses differential signaling through serial communication.

Sensing Systems

There are multiple sensing systems (shown in Table 1 below) within a vehicle that enables it to operate more safely with precision either autonomously or semi- autonomously and allow drivers to derive more information about the environment around them in addition to improving the occupants' comfort [3-5].

Table 1. Sensors in vehicle systems.

Sensor	Type	Usage	Function	Constraint
LiDAR	Environment	M-H	2D/3D Ambient Mapping	0 - 250 Meters
RADAR		M-H	Object Detection by Sound	30 - 250 Meters
Ultrasonic		M-H	Object Detection by Sound	0 - 2.5 Meters
GPS		L-M-H	Localization	Satellite Coverage
Camera		L-M-H	Ambience Imaging	Fog/Haze & Obstructions
Magnetic Encoders	Vehicle Dynamics	L-M-H	Control Through Detection of Mechanical Motion	Sensitive to Magnetic and Radio Interference
Inertial Sensors		L-M-H	ABS, Air Bag System, Electronic Static Control (ESC)	Drift Errors
TPMS		L-M-H	-	Inaccurate or False Readings

Data Acquisition and Analysis

Optimal cost and effectiveness were the key factors that were taken into consideration before carrying out data collection and analysis. For the hardware, a USB2CAN OBD-II reader from 8devices was used due to its low cost and off-the-shelf implementation. This hardware was interfaced with a virtual machine running Ubuntu 14.04. A Linux OS was chosen as there is a well-known and versatile CAN subsystem for Linux called SocketCAN that can read data directly from a vehicle's OBD-II port. The log is pre-processed to identify the 3 labeled headers (or features) i.e., time, payload content, and payload size.

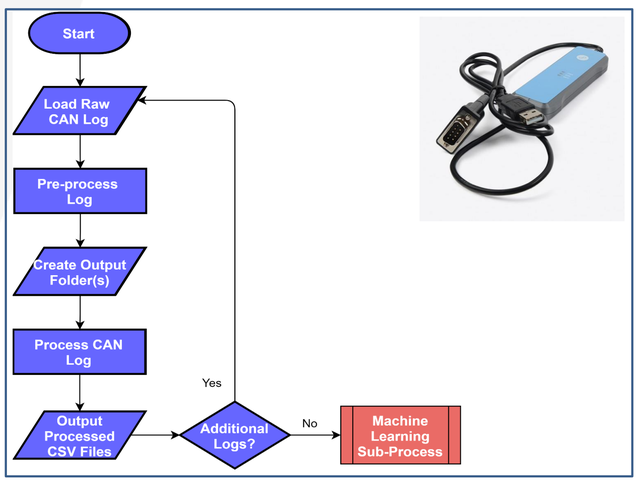

Figure 1: Flowchart - CAN data acquisition and processing.

Evaluation Metrics

Classification performance of the models were assessed using three evaluation metrics: F1-score (or F-measure), precision, and recall.

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{F1} - score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

where TP is true positive, FP is false positive, and FN is false negative.

Experimental Results

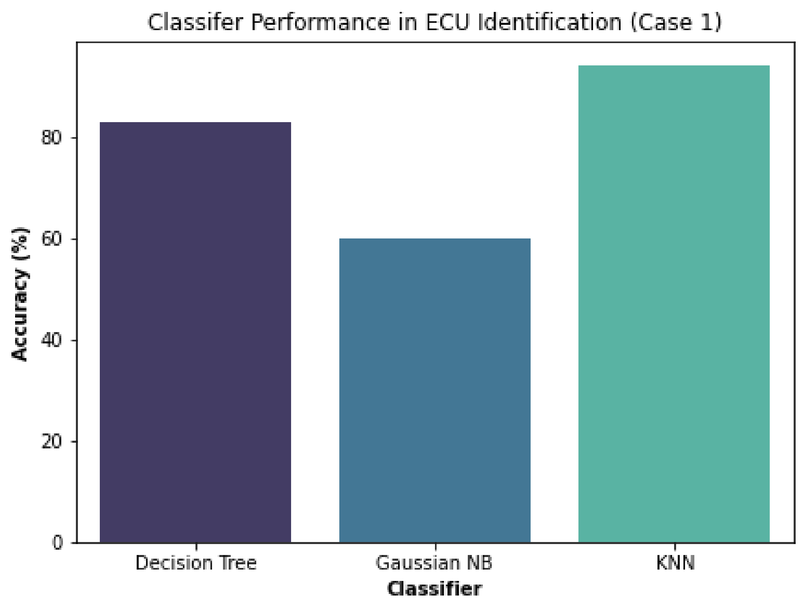

Table 2. Dataset details for scenario 1.

Class	No. of Samples	Function	Features	Split
Class 0	173,433	Brake	Time, Size, Data	60% - 40%
Class 1	346,785	Steering		
Class 2	346,918	Speed		
Class 3	677,763	Tachometer		
Class 4	72,296	Lighting		
Class 5	8,351,625	Other		

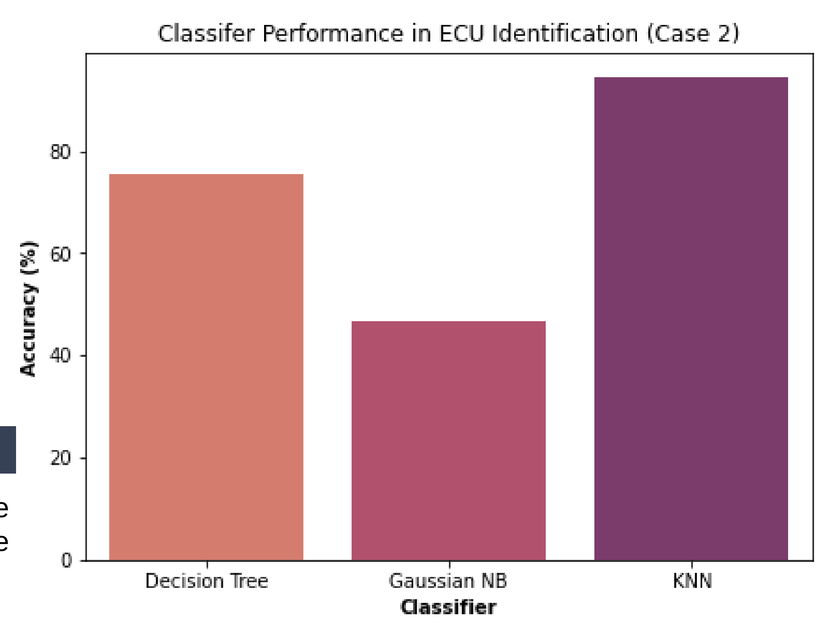

Figure 2. Classifier results for Decision Tree, Gaussian Naïve Bayes, and k-Nearest Neighbors for 2 test scenarios.

Table 3. Dataset details for scenario 2.

Class	No. of Samples	Function	Features	Split
Class 0	173,433	Brake	Time, Size, Data	60% - 40%
Class 1	346,785	Steering		
Class 2	346,918	Speed		
Class 3	677,763	Tachometer		
Class 4	72,296	Lighting		
Class 5	8,351,625	Other		
Class 6	60,036	Speed (Honda)		
Class 7	30,057	Speed (Toyota)		

Table 4. Model performances for test scenarios.

Accuracy	F1-score	Training Time	Testing Time	Model	Scenario
83.1%	0.83	76.7 ms	3.9 ms	DT	Scenario 1
94.3%	0.94	219 ms	1580 ms	kNN	
60.2%	0.61	28.9 ms	27.9 ms	GNB	
75.4%	0.74	128.6 ms	6.9 ms	DT	Scenario 2
94.5%	0.94	740 ms	2317 ms	kNN	
46.5%	0.39	89.4 ms	98.6 ms	GNB	

Conclusions

Rapid digitization of modern vehicles using electronic control units (ECUs) has made the modern automobile realize autonomous operations. ECUs within a vehicle is capable of handling multiple functions within the vehicle pertaining to vehicular control, infotainment system, or electronic control of mirrors, wipers, and seats. Such data are relayed through various communication buses like the controller area network (CAN). Manually identifying ECUs cannot work at scale due to labor-intensiveness. Therefore, to automate this process, three supervised machine learning algorithms (Decision Tree, Gaussian Naive Bayes, and kNN) were identified and compared for two datasets that are constructed differently based on vehicle makes. The classification algorithms show that the distance-based k-nearest neighbors (kNN) algorithm gives the highest performance, followed by the Decision Tree algorithm and the Gaussian Naive Bayes. Both these models generalize well enough during their training sets to produce accuracy and F1 scores that are higher than 70% and 0.65, respectively. These findings indicate that nonlinear models like kNN and Decision Tree show potential for identifying ECU signatures at scale.

Contact Information

Niroop Sugunaraj
Ph.D. Student
University of North Dakota
niroop.sugunaraj@und.edu

Acknowledgements

I'd like to acknowledge the support of my advisor Dr. Prakash Ranganathan in making this research work possible.

The findings in this poster were published in IEEE International Conference on Electro Information Technology (EIT), 2022. Please cite using:

Sugunaraj, N., & Ranganathan, P. (2022, May). Electronic Control Unit (ECU) Identification for Controller Area Networks (CAN) using Machine Learning. In 2022 IEEE International Conference on Electro Information Technology (eIT) (pp. 1-7). IEEE.

References

1. *R. Bosch, "Can specification version 2.0," Robert Bosch GmbH, Tech. Rep., 1991.*
2. *C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," Black Hat USA, vol. 2015, no. S 91, 2015.*
3. *O. Avatefipour, A. Hafeez, M. Tayyab, and H. Malik, "Linking received packet to the transmitter through physical-fingerprinting of controller area network," in 2017 IEEE Workshop on Information Forensics and Security (WIFS). IEEE, 2017, pp. 1–6.*
4. *M. Jaynes, R. Dantu, R. Varriale, and N. Evans, "Automating ecu identification for vehicle security," in 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA). IEEE, 2016, pp. 632–635.*
5. *K.-T. Cho and K. G. Shin, "Viden: Attacker identification on in-vehicle networks," in Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 2017, pp. 1109–1123.*

UNIVERSITY OF NORTH DAKOTA.
School of Electrical Engineering & Computer Science
UNIVERSITY OF NORTH DAKOTA