Georgia Institute of Technology

CS 4220/6235: RTES

Fall 2025

Project Checkpoint Report

Group: Solo
Name(s): Phineas Giegengack
Project: Use of Machine Learning to Detect Attacks on CAN (Controller Area Network) in Embedded Microcontroller

Project Plan (Scope):

Introduction

The Controller Area Network (CAN) protocol is a specification which describes how many electronic control units (ECUs) broadcast data over a shared bus. CAN was derived in the 1980s for use in automobiles to reduce the complexity of the electrical design of vehicles with an ever-growing array of electronic sensors and actuators. CAN was an answer to difficulties that arose from designing point-to-point connections between every node that needed to communicate in a car, but CAN does not by default have authentication, which leaves it open to a variety of attacks, including Denial of Service (DoS), Fuzzing, and Spoofing.

In modern vehicles, and especially in autonomous vehicles, safety-critical systems may be connected via CAN, so the ability to detect intrusions onto this network is of utmost importance.

Related Work

A variety of methods to detect attacks/intrusions on the CAN bus using machine learning have been proposed.

In [1], a traditional neural net and multi-layer perceptron are compared for their performance in detecting CAN bus intrusions. They are trained and tested on a static dataset of CAN bus message traces. No effort is made to optimize the models for implementation on an embedded device, but the models achieve respectable performance.

Researchers in [2] made use of a GRU model to achieve high performance with a lightweight model intended to be deployed on embedded hardware. They also propose a system architecture for deploying, updating and training the model over-the-air (OTA) which would be desired for real-world applications to keep model performance up to date as CAN bus attacks evolve.

[3] and [4] make use of support vector machines (SVMs) to do one-class classification which enables models to be trained only on ‘good’ CAN bus data and have the models infer when anomalous behavior is taking place without needing training data for those scenarios. This is

appealing since attack methods are likely unknown at the time of training, and having one-class models would allow detection of novel attacks before they impact any vehicle.

[5] is the most recent paper I was able to find and proposes another model type, TPE-LightGBM algorithm. They achieve good performance and the model is lightweight, although they do not deploy it on embedded hardware.

Methods

For this project, I will develop a lightweight machine learning solution for detecting anomalies in CAN communication that can be deployed on a low-cost microcontroller. I will select a machine learning technique for detecting CAN bus intrusions, select from a variety of open-source datasets of nominal and anomalous CAN bus activity, and select an embedded processor system for cost-effective performance. I will develop an architecture for training the model on static CAN bus traces and analyze a 'control model' that will run on my PC instead of an embedded processor. When the model is accurate relative to other models that have been developed in other papers, I will port the model to the embedded computing platform, experimenting with reducing resolution of weights, size of feature vector, and depth of neural network to make the model efficient to run in a real-time scenario while attempting to preserve performance where possible.

I hope to show a reliable method for training an ML model on CAN bus traces and deploying the trained model on an embedded computing platform to detect anomalous behavior on the CAN bus. I hope to compare model performance, model size, and speed of various iterations of the chosen machine learning technique to better understand trade-offs in deploying machine learning for real-time monitoring of the CAN bus.

Schedule

RTES 2025 GANTT CHART											
Date	9/25/2025		10/9/2025		10/23/2025		11/6/2025		11/20/2025		12/4/2025
Week	5	6	7	8	9	10	11	12	13	14	15
Ckpt	1		2		3		4		5		6
Tasks											
Background Reading											
Selection of Hardware											
Selection of Machine Learning Strategy											
Selection of Dataset(s)											
Programming Non-Embedded ML Model											
Analysis of Non-Embedded ML Model											
Hardware System Design											
Hardware Assembly											
Port ML Model to Embedded Platform											
Analysis of Embedded ML Model											
Final Report											

Current progress report (Match):

Checkpoint 1

In the past two weeks I have reviewed related works in the area of CAN bus intrusion detection with machine learning. I have read and taken notes on 5 papers which have given me a good foundation of knowledge to begin this project. I also created the github and project timeline for the project, so naturally I am on-track with the project so far.

In the next two weeks, my focus will be refining the project details to remove as many 'TBD's as possible. Namely, I will select the hardware platform, machine learning technique, and

dataset which will be used for the remainder of the project. The goal is to order any components I need as early as possible, so they are available for the rest of the semester. I will conduct risk-management for these selections by doing a 'hello world' demo using the selected machine learning technique and do some statistical analysis of the selected dataset to ensure they will be good choices for the project.

Supporting Evidence (Factual): • https://github.com/finncg1234/RTES-2025

See the github link above for the project timeline (Gantt Chart) and annotated bibliography. The annotated bibliography shows citations for 5 papers I read and my thoughts on how I can use the results they had to inform my project. These will be especially helpful for selecting a machine learning architecture to use during the next two weeks.

Skill Learning Report:

Research: I have read 5 papers, doing my best to absorb the most relevant information to my project. I have honed in on machine learning models and their tradeoffs for real-time systems.

Project Planning: I created a Gantt chart to organize my semester and keep me on-track to complete this project. I built in cushion time so that unexpected delays hopefully will not keep me from delivering a completed project.

Self-Evaluation:

Scope: 90%: I think the scope of this project is pretty large. At this time, I admit that there are a lot of unknowns in the methodology for this project, which counts against my score for Scope, but I acknowledge those and include the resolution of unknowns in my project plan (selection of hardware, machine learning method, dataset). I think this project is interesting, important, and relevant to real-time systems, so I believe I have earned a respectable score for scope.

Match: 85%: I gave myself a lower score for match since in my project plan, I said I would do *all* background reading for this project, but I am aware that I will need to do more research over the course of this project. Certainly, I could have read more papers, although the papers I chose gave me a really good foundation and understanding of the state-of-the-art in this space.

Factual: 100%: I make no false claims in my match section, and by following the link to the github, I think you will agree that I did indeed read the papers I referenced and that I have a good understanding of how their findings fit into my project plan.

References: (also available in annotated bibliography on github)

[1] F. Amato, L. Coppolino, F. Mercaldo, F. Moscato, R. Nardone, and A. Santone, "CAN-Bus Attack Detection With Deep Learning," IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 8, pp. 5081–5090, Aug. 2021, doi: 10.1109/TITS.2020.3046974.

[2] H. Ma, J. Cao, B. Mi, D. Huang, Y. Liu, and S. Li, “A GRU-Based Lightweight System for CAN Intrusion Detection in Real Time,” Security and Communication Networks, vol. 2022, pp. 1–11, 2022, doi: 10.1155/2022/5827056.

[3] J. Guidry, F. Sohrab, R. Gottumukkala, S. Katragadda, and M. Gabbouj, “One-Class Classification for Intrusion Detection on Vehicular Networks,” in Proc. 2023 IEEE Symposium Series on Computational Intelligence (SSCI), Mexico City, Mexico, 2023, pp. 1176–1182, doi: 10.1109/SSCI52147.2023.10371899.

[4] C. Chupong, N. Junhuathon, K. Kitwattana, T. Muankhaw, N. Ha-Upala, and M. Nawong, “Intrusion Detection in CAN Bus using the Entropy of Data and One-class Classification,” in Proc. 2024 International Conference on Power, Energy and Innovations (ICPEI), Nakhon Ratchasima, Thailand, 2024, pp. 157–160, doi: 10.1109/ICPEI61831.2024.10748816.

[5] L. Liang, Y. Zhao, H. Zhang, L. Chen, X. Wang, and Y. Li, “Intrusion Detection Model for In-vehicle CAN Bus Based on TPE-LightGBM Algorithm,” in Proc. 2025 IEEE 34th Wireless and Optical Communications Conference (WOCC), Taipa, Macao, 2025, pp. 419–423, doi: 10.1109/WOCC63563.2025.11082193.