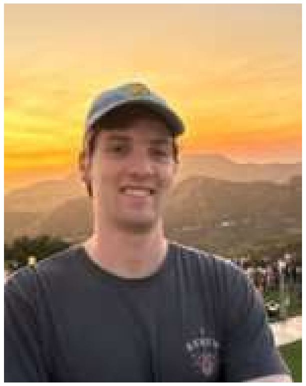
Machine Learning to Detect Attacks on CAN (Controller Area Network)

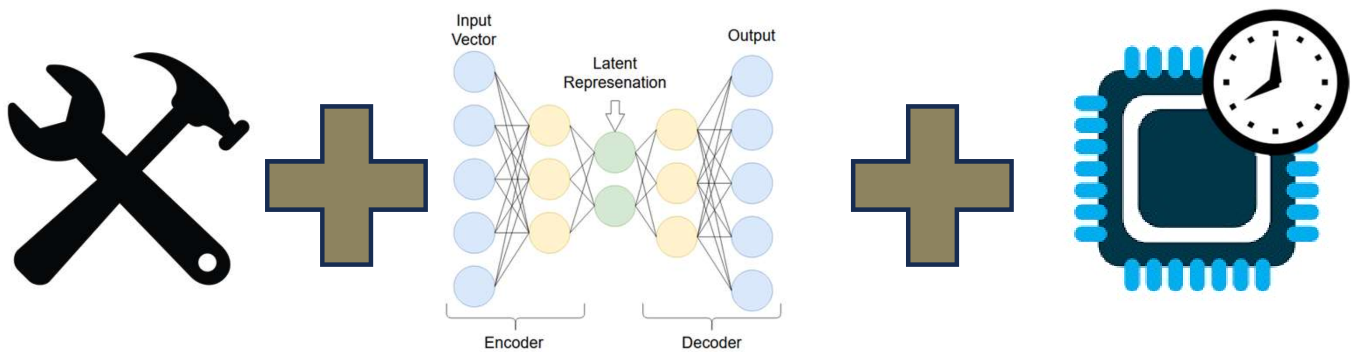


Finn Giegengack

Georgia Tech College of Engineering
School of Electrical and Computer Engineering

Project Overview

- Deliverables
  - Machine learning pipeline to detect attacks on the CAN bus
  - Develop an autoencoder which can detect most attacks on CAN bus with reasonable accuracy

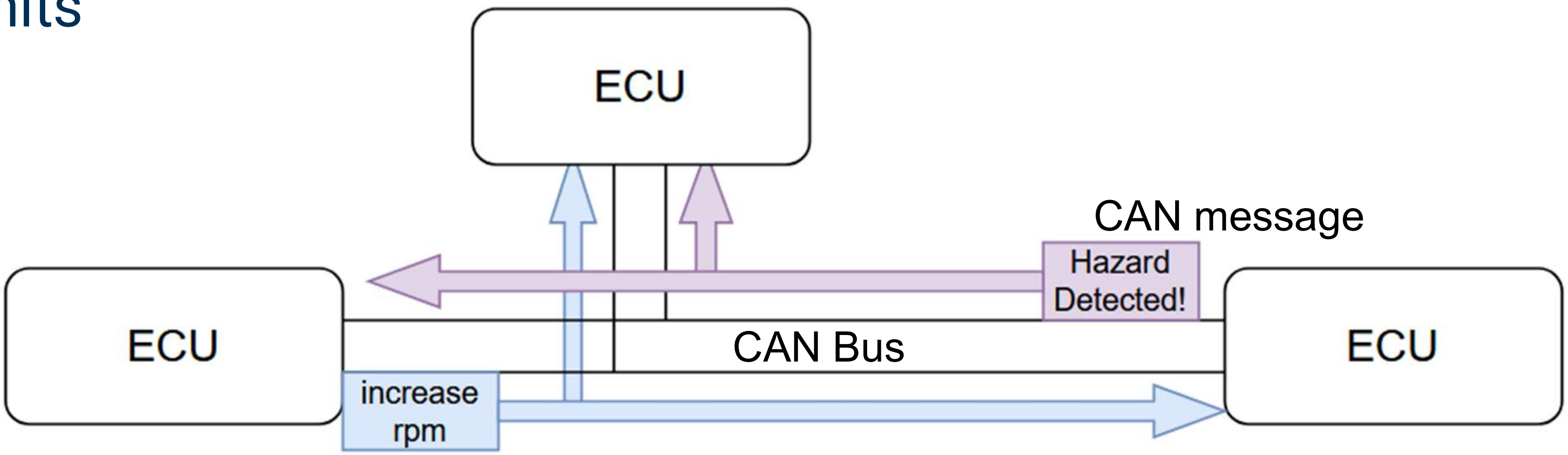


Project Overview

- Deliverables
  - Machine learning pipeline to detect attacks on the CAN bus
  - Develop an autoencoder which can detect most attacks on CAN bus with reasonable accuracy

Background | Methods | Results | What's Next?

- CAN protocol
- Used commonly in automobiles
- Coordinates Communication between multiple Electronic Control Units



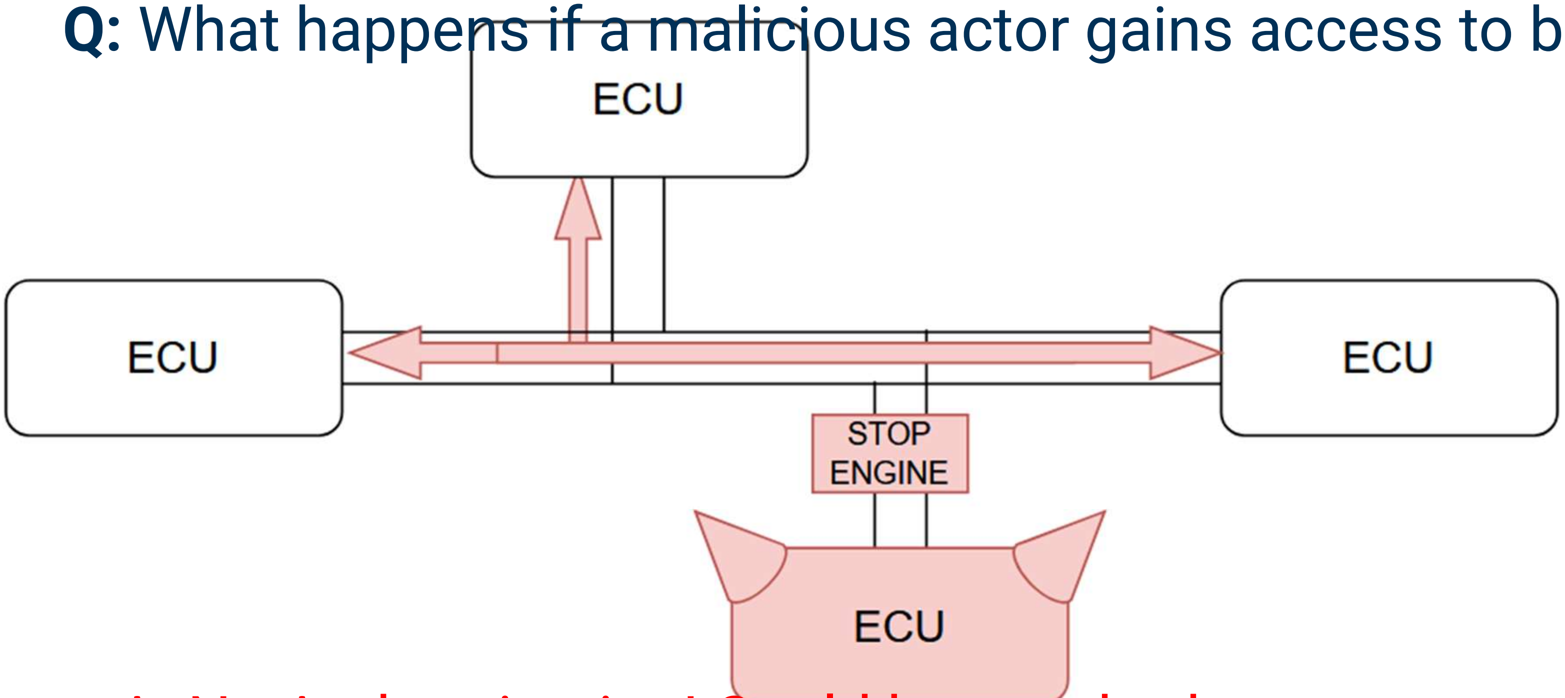
Background | Methods | Results | What's Next?

- Designed before wireless, over-the-air updates, …
- **Threat model/assumptions have changed**

Q: What happens if a malicious actor gains access to bus?

- Designed before the internet, over-the-air updates, …
- **Threat model/assumptions have changed**

Q: What happens if a malicious actor gains access to bus?



A: No Authentication! Could be very bad.

Background

Next?

TABLE I. SUMMARY OF EXPLOITS AGAINST THE CONTROLLER AREA NETWORK (CAN) BUS.

Vulnerability	***Attack Type***	***Result***
Unauthorized access	Packet injection. Reflashing ECU while driving.	Engine stopped. Code loaded into vehicles' telematics unit.
Unauthorized access Denial of Service	Packet injection to Body Control Module (BCM). Fuzzing.	Door lock relay activated. Wipers turned on/forced off. Trunk opened. Horn activated. Auto-headlight control deactivated. Use of washer fluid. Brake/auxiliary lights rendered inoperable.
Unauthorized access	Packet injection to Engine Control Module (ECM)	Engine timing and RPM disturbed. Engine cylinders stopped. Grind starter motor
Unauthorized access	Packet injection to Brake Control Module (BCM)	Brake application and release (evenly or unevenly) at speeds below 5 mph.
Denial of Service	Packet injection to other CAN-connected bus devices.	Disable CAN bus communication. Freeze instrument panel status.
Unauthorized access	Packet injection	Kill engine

[9] J. N. Brewer and G. Dimitoglou, "Evaluation of Attack Vectors and Risks in Automobiles and Road Infrastructure," in 2019 International Conference on Computational Science andComputational Intelligence (CSCI), Las Vegas, NV, USA: IEEE, Dec. 2019, pp. 84–89. isbn: 978-1-7281-5584-5. doi: 10.1109/CSCI49370.2019.00021. Accessed: Nov. 7, 2025. [Online].

Background | Methods | Results | What's Next?

- How can we detect attacks on the CAN bus using ML?
- Lots of related work…

2024 International Conference on Power, Energy and Innovations (ICPEI 2024)
October 16-18, 2024, Nakhon Ratchasima (Korat), THAILAND

Intrusion Detection in CAN Bus using the Entropy of Data and One-class Classification

Charnon Chupong* Electrical Engineering Dept. Rajamangala University of Technology Thanyaburi Pathum Thani, Thailand charnon.c@en.rmutt.ac.th

Nitikorn Junhuathon Electrical Engineering Dept. Rajamangala University of Technology Thanyaburi Pathum Thani, Thailand nitikorn.j@en.rmutt.ac.th

Krit Kitwattana Electrical Engineering Dept. Rajamangala University of Technology Thanyaburi Pathum Thani, Thailand krit.k@en.rmutt.ac.th

Hindawi
Security and Communication Networks
Volume 2022, Article ID 5827056, 11 pages
https://doi.org/10.1155/2022/5827056

WILEY | Hindawi

Research Article

A GRU-Based Lightweight System for CAN Intrusion Detection in Real Time

Haoyu Ma, Jianqiu Cao, Bo Mi, Darong Huang, Yang Liu, and Shaoqian Li
School of Information Science and Engine

One-Class Classification for Intrusion Detection on Vehicular Networks

…lry*, Fahad Sohrab†, Raju Gottumukkala* Satya Katragadda† and Moncef Gabbouj‡
*Mechanical Engineering, University of Louisiana at Lafayette, United States
Informatics Research Institute, University of Louisiana at Lafayette, United States
…y of Information Technology and Communication Sciences, Tampere University, Finland
…ke.guidry1@lousiana.edu, fahad.sohrab@tuni.fi, raju.gottumukkala@louisiana.edu, satyasivakumar@gmail.com, moncef.gabbouj@tuni.fi

…to external networks such as the Internet. These modern features open vehicles' networks to new vectors of attack that they were never designed for. This openness leaves vehicles as prime targets for well-known and widely used cyber-security attacks [2]–[4]. The gravity of these threats is exhibited by researchers who performed hacks on Tesla vehicles that disabled critical safety features [5]. As these threats become more apparent, automobile manufacturers are implementing countermeasures such as data encryption and…

SACI 2025 • IEEE 19th International Symposium on Applied Computational Intelligence and Informatics • May 19–24, 2025 • Budapest (HU); Timişoara (RO)

Autoencoder based CAN BUS IDS system architecture and performance evaluation

Benedek Máté Tóth
John von Neumann Faculty of Informatics
Óbuda University
Budapest, Hungary
tothbenedek@stud.uni-obuda.hu

Anna Bánáti
John von Neumann Faculty of Informatics
Óbuda University
Budapest, Hungary
banati.anna@nik.uni-obuda.hu

***Abstract*—With the increasing sophistication and frequency of cyber threats targeting modern vehicles [1], the need for efficient, real-time intrusion detection systems (IDS) has become critical. This study explores the feasibility of deploying an autoencoder-based IDS [2] for CAN BUS networks in resource-constrained automotive environments. Unlike conventional deep learning approaches that rely on computationally intensive models, this research focuses on developing a lightweight, optimized autoencoder architecture capable of detecting anomalies while…**

safety (e.g., adaptive cruise control, Internet access to car systems), but on the other hand poses new safety challenges. One of the oldest internal data communication system in cars is CAN BUS, which is one of the most vulnerable ones. Robert Bosch developed the CAN BUS system in 1986 to replace point-to-point systems. CAN BUS is a much simpler, more robust, and fault-tolerant system, which is particularly critical in the automotive environment (CAN BUS is not only…

CAN-Bus Attack Detection With Deep Learning

Flora Amato, Luigi Coppolino, Francesco Mercaldo, Francesco Moscato, Roberto Nardone, and Antonella Santone

***Abstract*—Modern cars include a huge number of sensors and actuators, which continuously exchange data and control commands. The most used protocol for communication of different components in automotive system is the Controller Area Network (CAN). According to CAN, components communicate by broadcasting messages on a bus. In addition, the standard definition of the protocol does not provide information for authentication, so exposing it to attacks. This paper proposes a method based on deep learning aiming at discovering attacks towards the CAN-bus. In particular, Neural Networks and MultiLayer Perceptrons are the class of networks employed in our approach. We also validate our approach by analysing a real-world dataset with the injection of messages from different types of attacks: denial of service, fuzzy pattern attacks, and attacks against specific components. The obtained results are encouraging and demonstrate the effectiveness of the approach.**

Index Terms—Automotive, intelligent systems, deep learning, neural networks, attack detection, security.

I. Introduction

In the last years, the automotive domain has experienced a rapid growth of complexity, mainly due to the introduction of more and more complex components for control, super-…

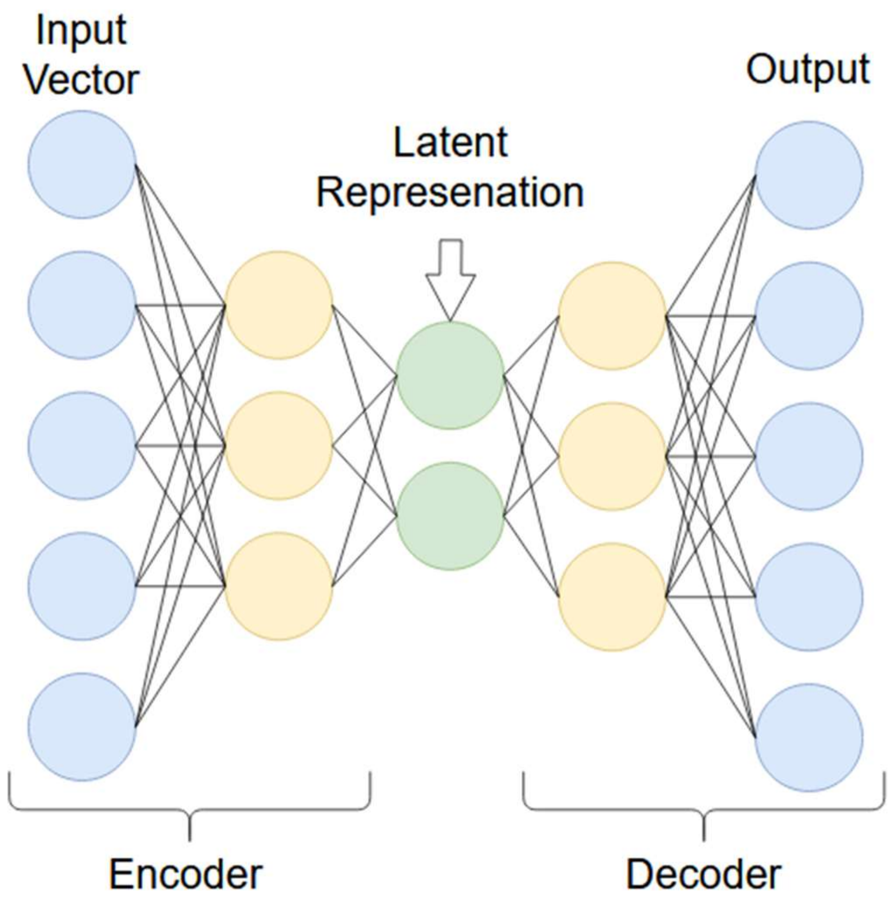
ECUs execute many safety controls, such as skid detection, crash prediction or anti-lock braking [1]. In general, the whole control system, so considerably changed during the years, has soft or hard on real-time requirements on network components in order to guarantee the correct operation and the sufficient quality levels (i.e., performance, reliability, safety) of the whole system.

The most used communication standard is the Controller Area Network (CAN) (also known as CAN-bus) [3]. The CAN standard defines a communication protocol; it dates back to the 80s and it was developed by Bosch GmbH company in order to provide efficient communication among several automotive applications, also with a critical impact on the safety. It became a standard within the ISO 11898-1:2011.[1] The CAN standard was initially applied to automotive, but it is currently employed in many industrial applications for communication of embedded components. It mainly consists in a broadcast protocol where messages sent by a sender are analysed by all receivers connected to the same network, which in turn decide if the message have to be delivered at application layer [4].

Georgia Tech College of Engineering
School of Electrical and Computer Engineering

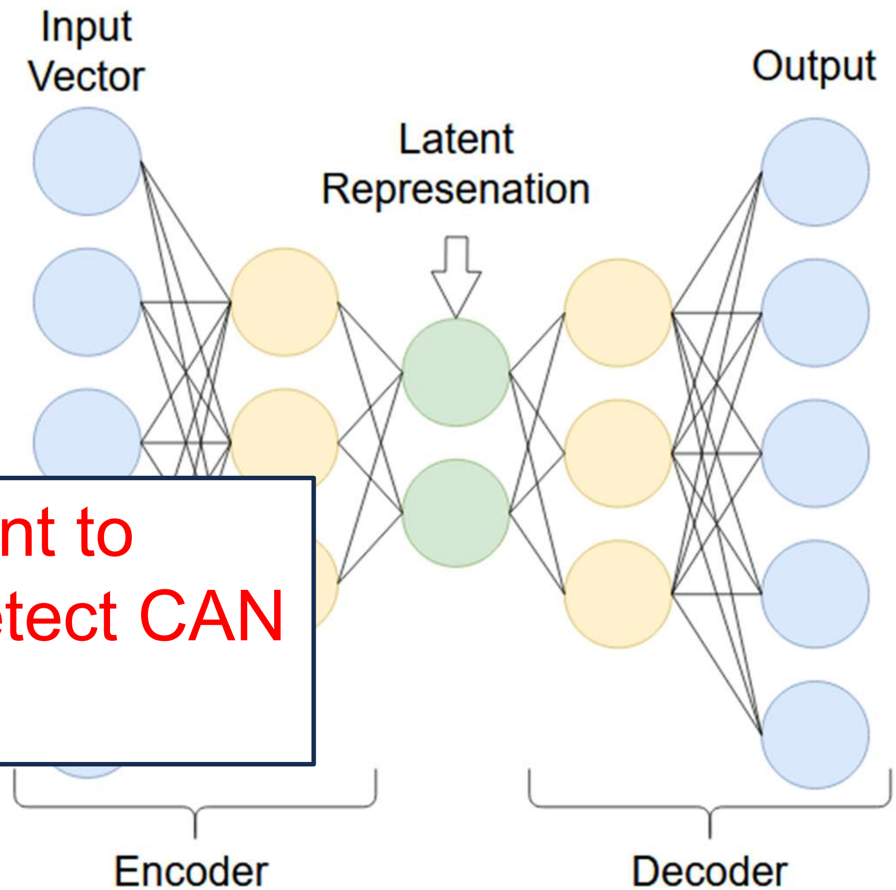
Background | Methods | Results | What's Next?

- What is an autoencoder?
- Learns to 'reconstruct' input



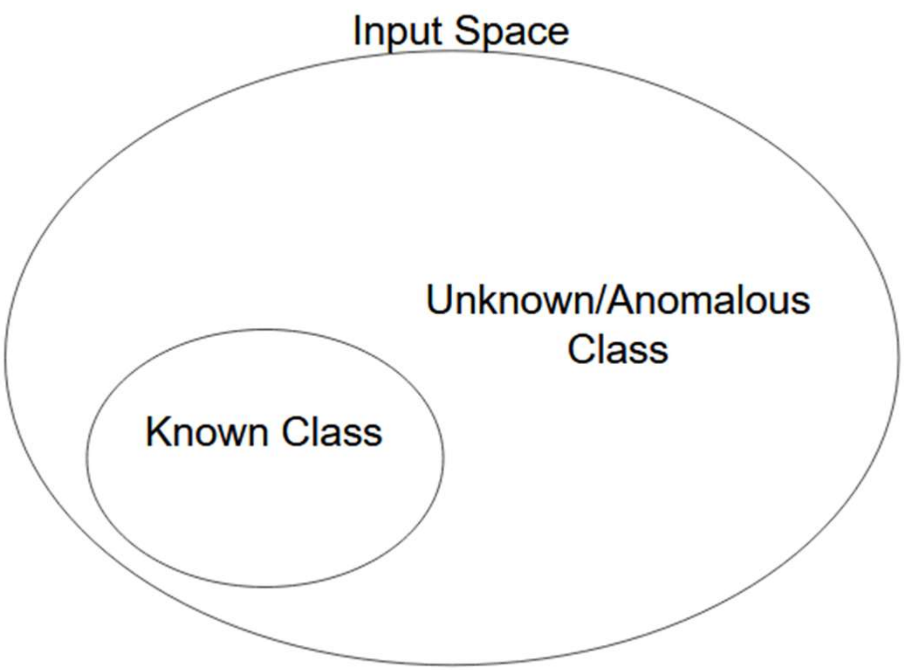
Background | Methods | Results | What's Next?

- What is an autoencoder?
- Learns to 'reconstruct' input



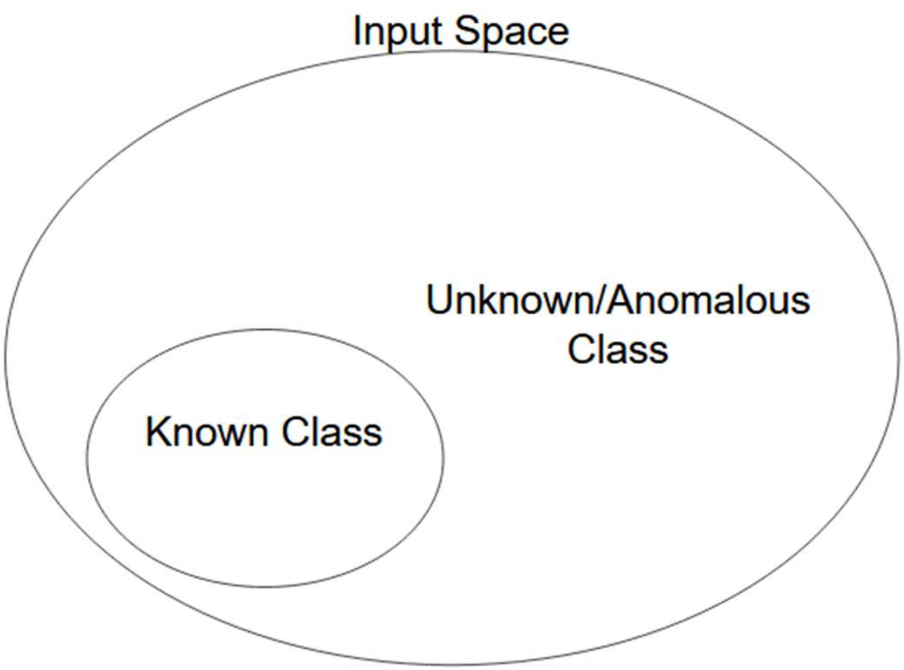
But why would we want to reconstruct data to detect CAN bus attacks?

- One Class Classification
  - Subverts challenges with detecting CAN bus attacks: lack of attack data



- We would like to train only on known data and detect anomalies

- One Class Classification
  - Subverts challenges with detecting CAN bus attacks: lack of attack data



- We would like to train only on known data and detect anomalies
- **Train only on Attack-Free CAN Bus DATA, detect attacks, which fall outside of known class**

Background | **Methods** | Results | What's Next?

- One Class Classification with Autoencoder
  - Train autoencoder on 'good' attack-free CAN bus data
  - Select reconstruction error threshold to maximize true positives, minimize false positives.
  - Challenges:
    - Autoencoder is characteristically good at generalizing to unseen inputs, but in this case that behavior is undesirable.
  - Possible Solutions:
    - Select latent dimension that is sufficiently small to promote overfitting on known class
    - **Use of Batch-Wise Feature Weighting block (BFW) as described in [6]**

[6] Y. Liao and B. Yang, "To Generalize or Not to Generalize: Towards Autoencoders in One-Class Classification," in 2022 International Joint Conference on Neural Networks (IJCNN), Padua, Italy: IEEE, Jul. 18, 2022, pp. 1–8, isbn: 978-1-7281-8671-9. doi: 10.1109/IJCNN55064.2022.9892812. Accessed: Nov. 7, 2025. [Online].

Background | **Methods** | Results | What's Next?

- Dataset
  - can-train-and-test [8]
  - Contains attack-free CAN bus DATA and attack scenarios

```
autoencoder_can > data > 2016-chevrolet-silverado-DoS-labeled.csv
16240
16241
16242    timestamp,arbitration_id,data_field,attack
16243    1672531205.964396,000,0000000000000000,1
16244    1672531205.965094,1ED,6140020801B870DE,0
16245    1672531205.965409,000,0000000000000000,1
16246    1672531205.9661832,182,98C8010000000000,0
16247    1672531205.966197,2F9,1B010A0800003B,0
16248    1672531205.966221,000,0000000000000000,1
```

DoS Attack

8] B. Lampe and W. Meng, "Can-train-and-test: A New CAN Intrusion Detection Dataset," in 2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall), Hong Kong, Hong Kong: IEEE, Oct. 10, 2023, pp. 1–7, isbn: 979-8-3503-2928-5. doi: 10.1109/VTC2023-Fall60731.2023.10333756. Accessed: Nov. 7, 2025. [Online]. Available: https://ieeexplore.ieee.

Background | **Methods** | Results | What's Next?

- Dataset
  - can-train-and-test [8]
  - Contains attack-free CAN bus DATA and attack scenarios

dataset_type	total_messages	normal_messages	attack_messages	attack_percentage	duration_seconds	messages_per_second	unique_can_ids
attack-free	1254632	1254632	0	0	487.7283158	2572.399345	98
combined	920226	918266	1960	0.212991157	357.0358989	2577.404689	98
dos	252707	208924	43783	17.32559842	81.25668597	3109.984083	99
fuzzy	1059035	1008513	50522	4.770569433	392.2755001	2699.722516	2048
gear	1257602	1254632	2970	0.236163747	487.728138	2578.489741	98
interval	950657	943319	7338	0.771887232	366.811969	2591.673883	98
rpm	524212	522688	1524	0.290722074	203.2086239	2579.673982	98
speed	1255235	1254632	603	0.048038813	487.727381	2573.640622	98
standstill	524245	522688	1557	0.296998541	203.2089219	2579.832593	98

8] B. Lampe and W. Meng, "Can-train-and-test: A New CAN Intrusion Detection Dataset," in 2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall), Hong Kong, Hong Kong: IEEE, Oct. 10, 2023, pp. 1–7, isbn: 979-8-3503-2928-5. doi: 10.1109/VTC2023-Fall60731.2023.10333756. Accessed: Nov. 7, 2025. [Online]. Available: https://ieeexplore.ieee.

Background | **Methods** | Results | What's Next?

- Feature Extraction
  - Take next N messages
  - Timestamp for sample 1 = 0
    - Timestamp for subsequent messages, n = timestamp_n – timestamp_1
  - Pack timestamp, CAN ID, and (optionally) CAN Data broken up by byte for all N messages into a single input vector
- Consideration:
  - Ignore CAN Data Field
  - Other feature extraction methods described in [4] involve extracting density of certain messages in moving window.

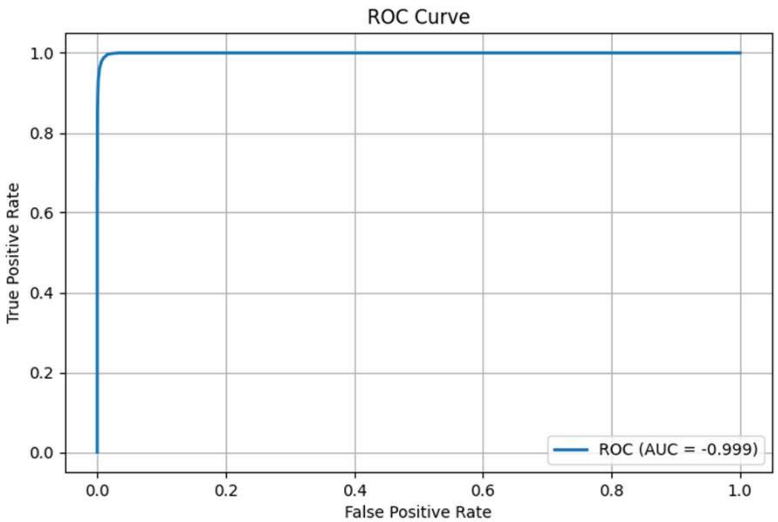
[4] C. Chupong, N. Junhuathon, K. Kitwattana, T. Muankhaw, N. Ha-Upala, and M. Na-wong, "Intrusion Detection in CAN Bus using the Entropy of Data and One-class Classification," in 2024 International Conference on Power, Energy and Innovations (ICPEI), Nakhon Ratchasima, Thailand: IEEE, Oct. 16, 2024, pp. 157–160, isbn: 979-83503-5677-9.doi: 10 . 1109 / ICPEI61831 . 2024 . 10748816. Accessed: Nov. 7, 2025.

Background | Methods | Results | What's Next?

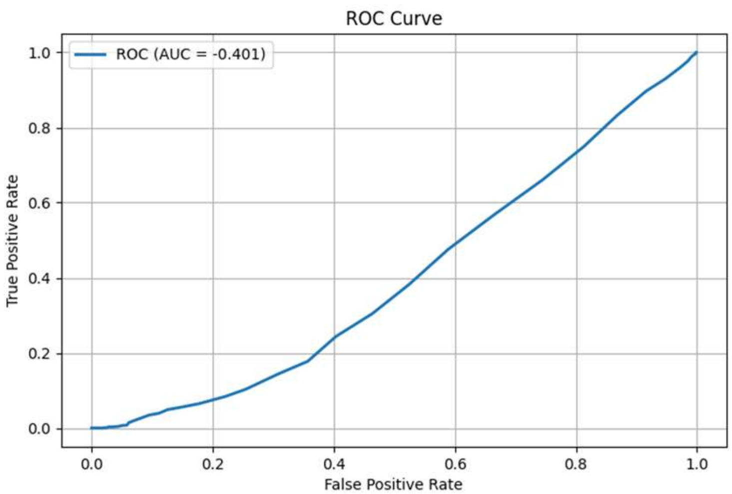
- Evaluation
  - can-train-and-test includes 8 attack scenarios to test against
  - DoS, fuzzing
  - gear, interval, rpm, speed, standstill
    - involve spamming CAN message with known malicious effect, e.g. increasing rpm

Background | Methods | Results | What's Next?

- Analysis – ROC (Receiver Operating Characteristic) curve and AUC (Area Under Curve)
  - For one class classification, want to maximize the AUC of the ROC
  - ROC: plots true positive rate vs false positive rate for different threshold selections



Outstanding

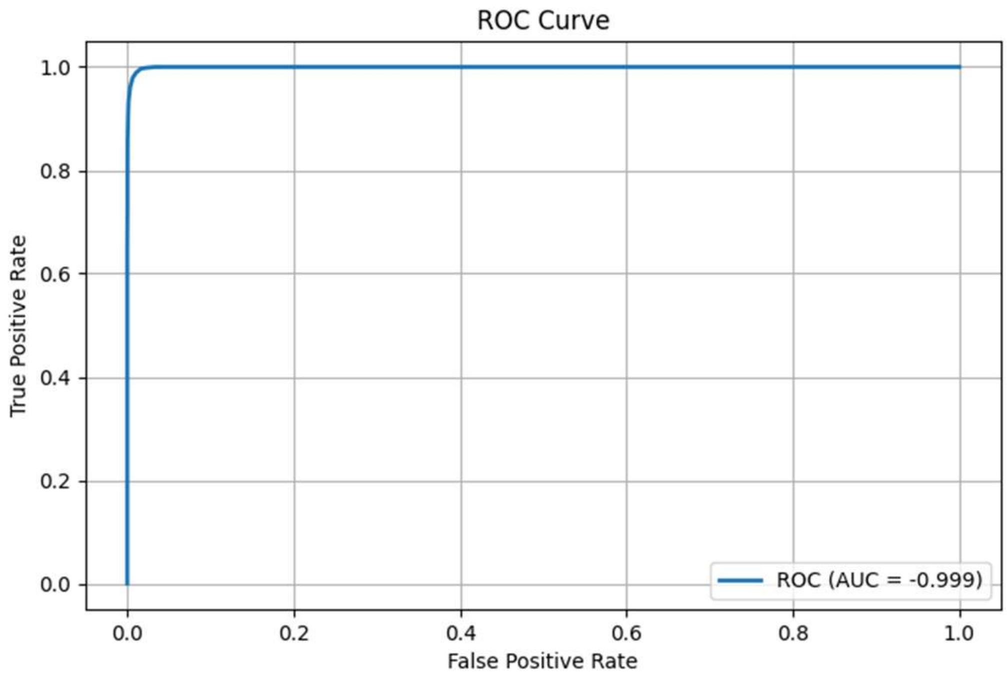


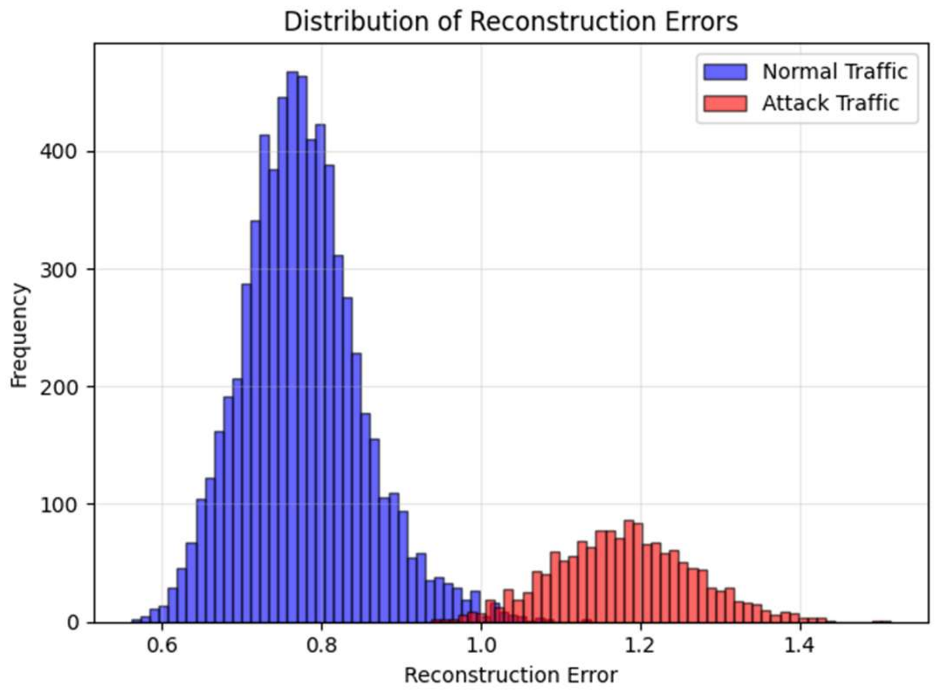
Terrible

Georgia Tech College of Engineering
School of Electrical and Computer Engineering

Background | Methods | Results | What's Next?

- Analysis – ROC Curve, another look

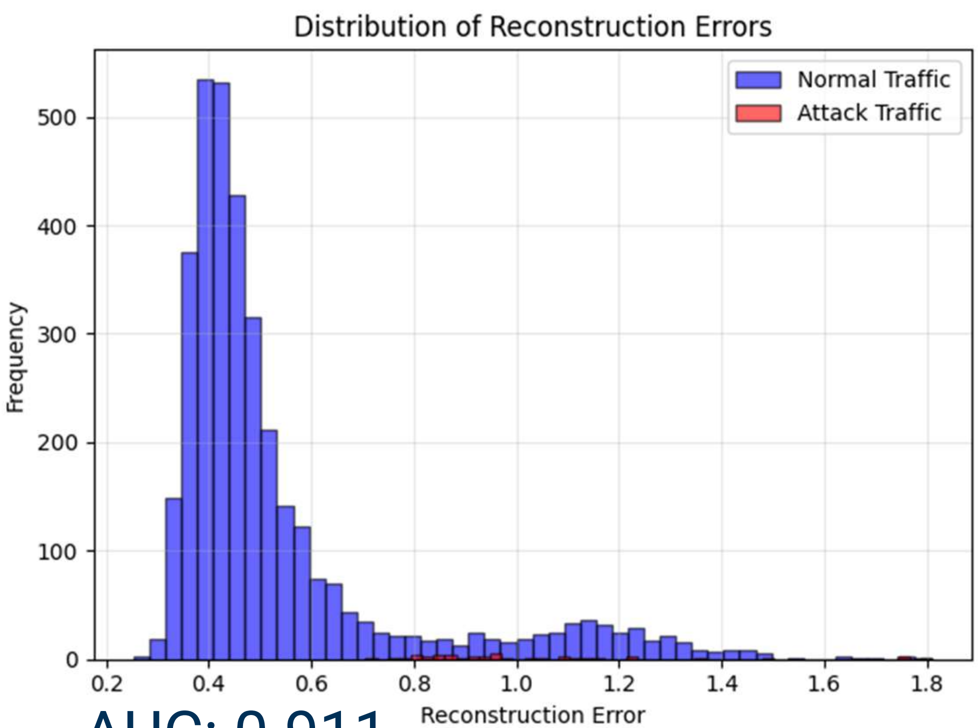


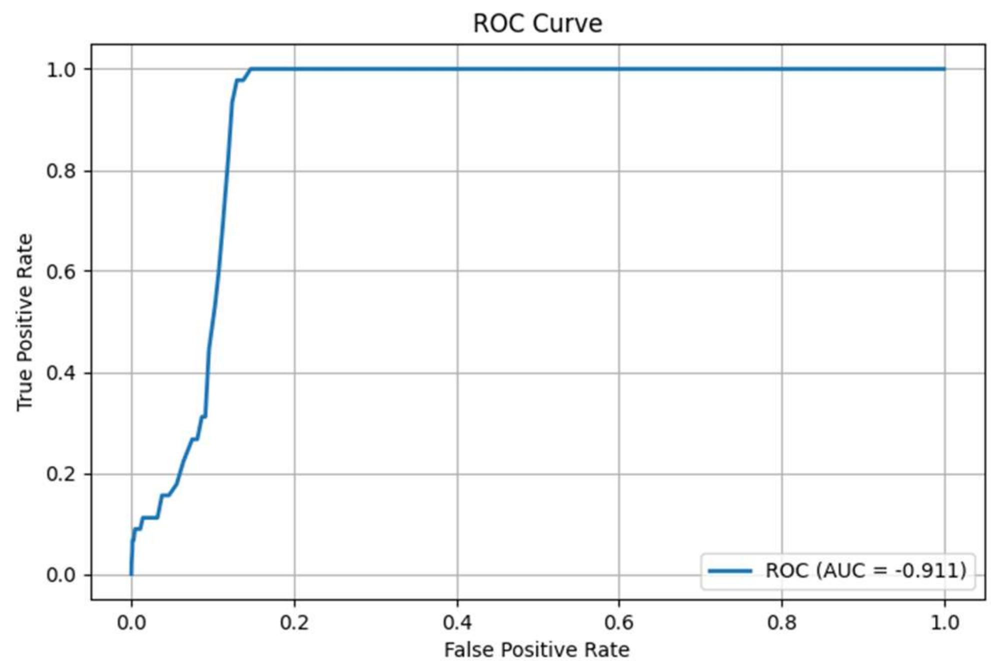


- Finding Optimal Parameters
  - N: number of CAN messages in a single input vector (size of window)
  - Learning rate
  - Batch size
  - Number of epochs
  - Feature extraction: whether to include CAN Data

Background | Methods | Results | What's Next?

- Best results for Combined Attacks





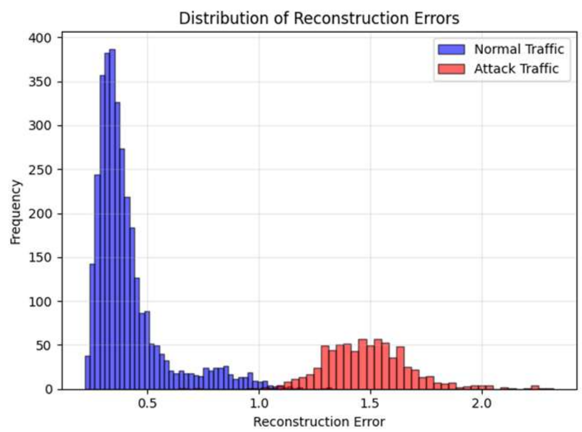
AUC: 0.911

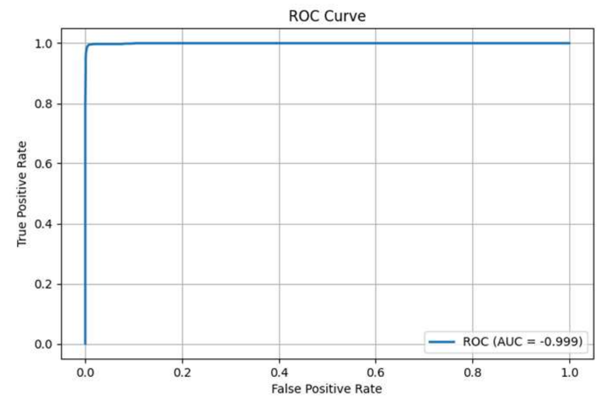
N=256, lr=0.0005, 2000 epochs, ignore CAN data

Georgia Tech College of Engineering
School of Electrical and Computer Engineering

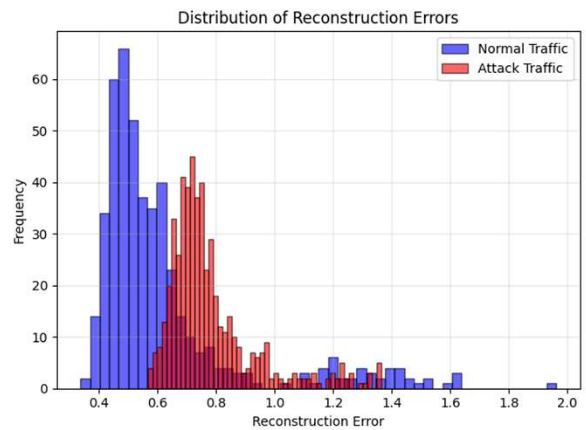
Background | Methods | **Results** | What’s Next?

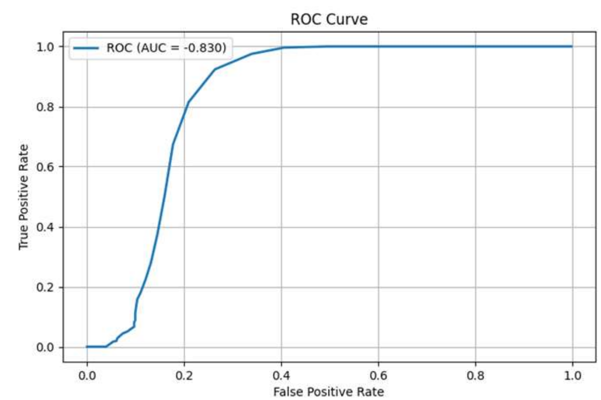
- Fuzzing Attacks





- DoS Attacks
  - Harder to predict?





- But!
- F1 Scores tell a different story

attack type	batch size	learning rate	epochs	msg per input	bfw	auc	max f1
combined	512	0.002	1000	256	1	0.932146771	0.188679245
dos	256	0.0018	1000	224	0	0.859448161	0.912710567
fuzzy	512	0.002	1000	224	0	0.999173709	0.998822144
gear	512	0.002	1000	256	1	0.90093119	0.215809285
interval	512	0.0001	1000	256	1	0.190671031	0.993494172
rpm	512	0.002	1000	256	1	0.906106613	0.275280899
speed	512	0.0005	1000	256	1	0.857480673	0.118012422
standstill	512	0.002	1000	256	1	0.906834394	0.277777778

- Effect of BFW on AUC

attack type	Average AUC w/ BFW	Average AUC w/o BFW	Difference in AUC mean	AUC Mean T Test p-value	Percent Improvement in AUC mean with BFW
combined	0.826095762	0.816539836	0.009555926	0.761548365	1.170295157
dos	0.80550158	0.814371122	-0.008869542	0.714614287	-1.089127731
fuzzy	0.977799241	0.977122082	0.000677159	0.97296619	0.069301324
gear	0.683509097	0.657284032	0.026225065	0.477526173	3.989913629
interval	0.572302369	0.564028981	0.008273388	0.84092411	1.466837379
rpm	0.79943465	0.785840837	0.013593813	0.647115285	1.72984298
speed	0.763801429	0.759404113	0.004397316	0.804667648	0.579048183
standstill	0.797168415	0.786118394	0.011050021	0.702841116	1.40564338

- Effect of BFW on F1 Score

attack type	Average F1 with BFW	Average F1 without BFW	Difference in F1 mean	F1 Mean T Test p-value	Percent Improvement in AUC mean with BFW
combined	0.117979391	0.111959004	0.006020388	0.590715696	5.37731434
dos	0.864217452	0.872967772	-0.00875032	0.591450952	-1.002364591
fuzzy	0.945836447	0.944790223	0.001046225	0.977353242	0.110736176
gear	0.086892499	0.078125988	0.008766511	0.378618931	11.22099173
interval	0.793008022	0.786842447	0.006165575	0.911550916	0.783584394
rpm	0.161738164	0.152070173	0.009667991	0.484399436	6.357585246
speed	0.075177555	0.073300329	0.001877225	0.689016663	2.561005362
standstill	0.162079824	0.155605887	0.006473938	0.644390019	4.160470943

Background | Methods | Results | **What's Next?**

- Try changing different parameters to improve F1
  - New feature extraction methods
  - Change latent dimension, dimensions of BFW
- Port model to Arduino Uno Q
- Explore other state of the art improvements to autoencoder for OCC

Thank You

References

[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, issn: 1524-9050, 1558-0016. doi: 10.1109/TITS.2020.3046974. Accessed: Nov. 7, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/9325944/.

[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, C. Chen, Ed., pp. 1–11, Jun. 27, 2022, issn: 1939-0122, 1939-0114. doi: 10.1155/2022/5827056. Accessed: Nov. 7, 2025. [Online]. Available: https://www.hindawi.com/journals/scn/2022/5827056/.

[3] J. Guidry, F. Sohrab, R. Gottumukkala, S. Katragadda, and M. Gabbouj, “One-Class Classification for Intrusion Detection on Vehicular Networks,” in 2023 IEEE Symposium Series on Computational Intelligence (SSCI), Mexico City, Mexico: IEEE, Dec. 5, 2023, pp. 1176–1182, isbn: 978-1-6654-3065-4. doi: 10.1109/SSCI52147.2023.10371899. Accessed: Nov. 7, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/10371899/.

References (Contd.)

[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 2024 International Conference on Power, Energy and Innovations (ICPEI), Nakhon Ratchasima, Thailand: IEEE, Oct. 16, 2024, pp. 157–160, isbn: 979-8-3503-5677-9. doi: 10 . 1109 / ICPEI61831 . 2024 . 10748816. Accessed: Nov. 7, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/10748816/.

[5] L. Liang et al., “Intrusion Detection Model for In-vehicle CAN Bus Based on TPE-LightGBM Algorithm,” in 2025 IEEE 34th Wireless and Optical Communications Conference (WOCC), Taipa, Macao: IEEE, May 20, 2025, pp. 419–423, isbn: 979-8-3315-3928-3. doi: 10.1109/WOCC63563.2025.11082193. Accessed: Nov. 7, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/11082193/.

[6] Y. Liao and B. Yang, “To Generalize or Not to Generalize: Towards Autoencoders in One-Class Classification,” in 2022 International Joint Conference on Neural Networks (IJCNN), Padua, Italy: IEEE, Jul. 18, 2022, pp. 1–8, isbn: 978-1-7281-8671-9. doi: 10.1109/IJCNN55064.2022.9892812. Accessed: Nov. 7, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/9892812/.

[7] B. M. T´oth and A. B´an´ati, “Autoencoder Based CAN BUS IDS System Architecture and Performance Evaluation,” in 2025 IEEE 19th International Symposium on Applied Computational Intelligence and Informatics (SACI), Timisoara, Romania: IEEE, May 19, 2025, pp. 000 099–000 104, isbn: 979-8-3315-1547-8. doi: 10 . 1109 / SACI66288 . 2025 . 11030168. Accessed: Nov. 7, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/11030168/

Georgia Tech College of Engineering
School of Electrical and Computer Engineering

References (Contd.)

[8] B. Lampe and W. Meng, “Can-train-and-test: A New CAN Intrusion Detection Dataset,” in 2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall), Hong Kong, Hong Kong: IEEE, Oct. 10, 2023, pp. 1–7, isbn: 979-8-3503-2928-5. doi: 10.1109/VTC2023-Fall60731. 2023.10333756. Accessed: Nov. 7, 2025. [Online]. Available: https://ieeexplore.ieee. org/document/10333756/.

[9] J. N. Brewer and G. Dimitoglou, “Evaluation of Attack Vectors and Risks in Automobiles and Road Infrastructure,” in 2019 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA: IEEE, Dec. 2019, pp. 84–89, isbn: 978-1-7281-5584-5. doi: 10.1109/CSCI49370.2019.00021. Accessed: Nov. 7, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/9071073/.

[10] M. Kemmler, E. Rodner, E.-S. Wacker, and J. Denzler, “One-class classification with Gaussian processes,” Pattern Recognition, vol. 46, no. 12, pp. 3507–3518, Dec. 2013, issn: 00313203. doi: 10.1016/j.patcog.2013.06.005. Accessed: Nov. 7, 2025. [Online]. Available: https: //linkinghub.elsevier.com/retrieve/pii/S0031320313002574

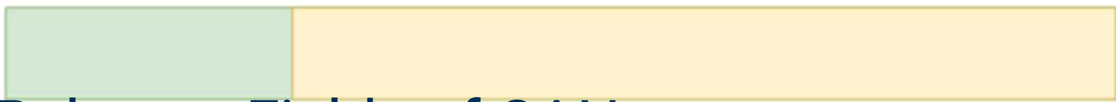
References (Contd.)

[11] J. Lee, S. Park, S. Shin, H. Im, J. Lee, and S. Lee, “ASIC Design for Real-Time CAN-Bus Intrusion Detection and Prevention System Using Random Forest,” IEEE Access, vol. 13, pp. 129 856–129 869, 2025, issn: 2169-3536. doi: 10.1109/ACCESS.2025.3585956. Accessed: Nov. 7, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/11071663/

[12] F. Sohrab, J. Raitoharju, M. Gabbouj, and A. Iosifidis, “Subspace Support Vector Data Description,” in 2018 24th International Conference on Pattern Recognition (ICPR), Beijing: IEEE, Aug. 2018, pp. 722–727, isbn: 978-1-5386-3788-3. doi: 10.1109/ICPR.2018.8545819. Accessed: Nov. 7, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/8545819/.

Appendix A: CAN Bus Message Description

- Relevant Fields of CAN message

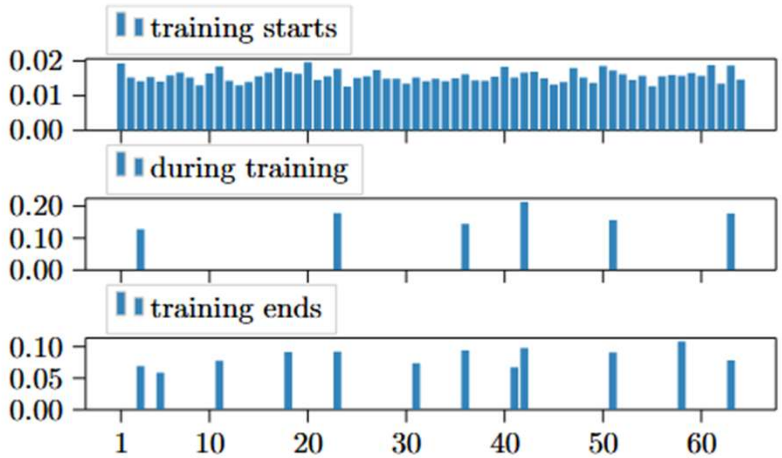


- can-train-and-test gives us
  - CAN ID, CAN data
  - Timestamp (time in seconds)
  - Label (attack or non-attack)

8] B. Lampe and W. Meng, “Can-train-and-test: A New CAN Intrusion Detection Dataset,” in 2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall), Hong Kong, Hong Kong: IEEE, Oct. 10, 2023, pp. 1–7, isbn: 979-8-3503-2928-5. doi: 10.1109/VTC2023-Fall60731.2023.10333756. Accessed: Nov. 7, 2025. [Online]. Available: https://ieeexplore.ieee.

Appendix B: Batch-Wise Feature Weighting

- Batch-Wise Feature Weighting (BFW)
  - Counteracts inherent ability of Autoencoder to generalize to new inputs
  - Assigns weighting to every value in latent dimension
  - Captures ‘most important’ latent features for known class, weighting others less



- Example of BFW training from Liao & Yang, 2022 [6]

[6] Y. Liao and B. Yang, “To Generalize or Not to Generalize: Towards Autoencoders in One-Class Classification,” in 2022 International Joint Conference on Neural Networks (IJCNN), Padua, Italy: IEEE, Jul. 18, 2022, pp. 1–8, isbn: 978-1-7281-8671-9. doi: 10.1109/IJCNN55064.2022.9892812. Accessed: Nov. 7, 2025. [Online].

Appendix B: Batch-Wise Feature Weighting

- Batch-Wise Feature Weighting (BFW)

```python
# BFW: 2-layer network to compute w from z
self.W1 = nn.Linear(latent_dim, bfw_hidden_dim)
self.W2 = nn.Linear(bfw_hidden_dim, latent_dim)

self.b1 = nn.Parameter(torch.zeros(bfw_hidden_dim))
self.b2 = nn.Parameter(torch.zeros(latent_dim))
```

- Trainable Parameters

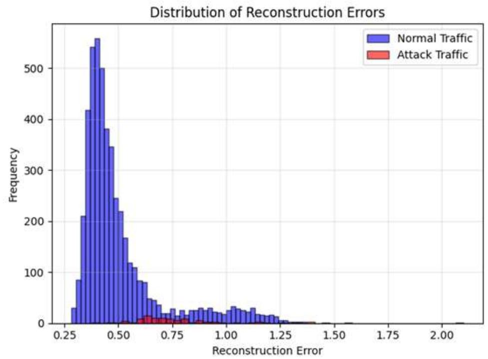
```python
def forward(self, x):
    # Encode to latent representation z
    z = self.encoder(x)

    if self.use_bfw:
        inner_bfw = nn.functional.relu(self.W1(z) + self.b1)
        outer_bfw = self.W2(inner_bfw) + self.b2
        batch_mean = outer_bfw.mean(dim=0, keepdim=True)
        w = nn.functional.softmax(batch_mean, dim=1)
        z = z * w

    decoded = self.decoder(z)
    return decoded
```

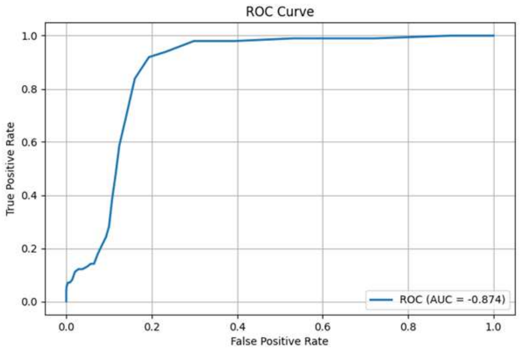
- Forward Pass applies BFW to latent representation

[6] Y. Liao and B. Yang, "To Generalize or Not to Generalize: Towards Autoencoders in One-Class Classification," in 2022 International Joint Conference on Neural Networks (IJCNN), Padua, Italy: IEEE, Jul. 18, 2022, pp. 1–8, isbn: 978-1-7281-8671-9. doi: 10.1109/IJCNN55064.2022.9892812. Accessed: Nov. 7, 2025. [Online].

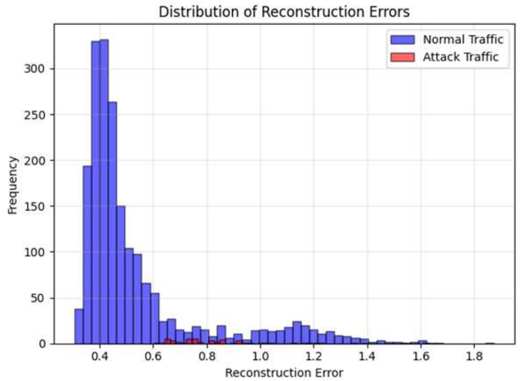
Appendix C: Results for Other Attacks

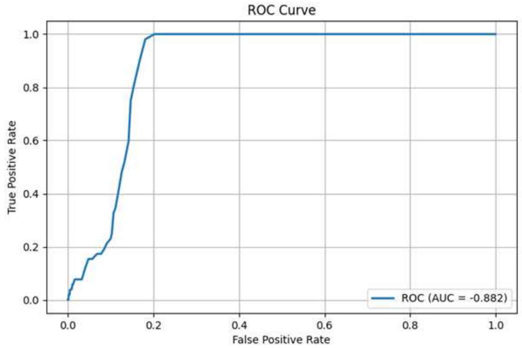
- Gear attack





- Rpm attack





Georgia Tech College of Engineering
School of Electrical and Computer Engineering