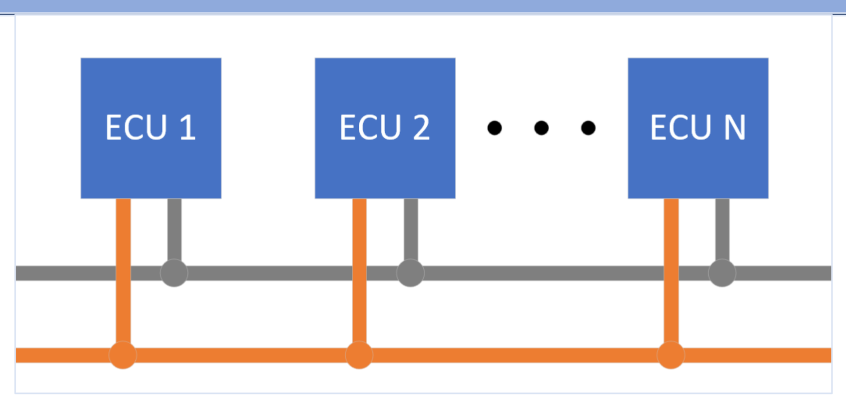
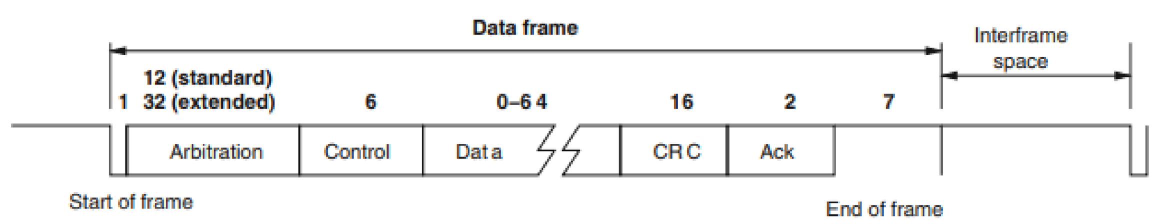
# Hybrid Anomaly Detection System to Prevent Malicious Attacks on Automotive CAN Networks

Jonathan Cochran (cochra48@purdue.edu), Logan Coles (colesl@purdue.edu), and John Mushatt (jmushatt@purdue.edu)

## Introduction




- What is CAN?
  - Controller Area Network protocol
    1. Developed in 1983 and widely used in the automotive industry
    2. Nodes are connected and communicate via a bus
       - Bus acts as a wired AND channel
- What is an ECU? [1]
  - An ECU is an electronic control unit and often used interchangeably with a node in CAN




- Start of frame: denotes the start of a CAN frame
- Arbitration: used for identifying message priority
- Control: defines how long the data payload is
- Data: payload a node wishes to send
- CRC: cyclic redundancy check for error detection
- Ack: acknowledgement for receiving messages
- End of frame: marks the end of a message

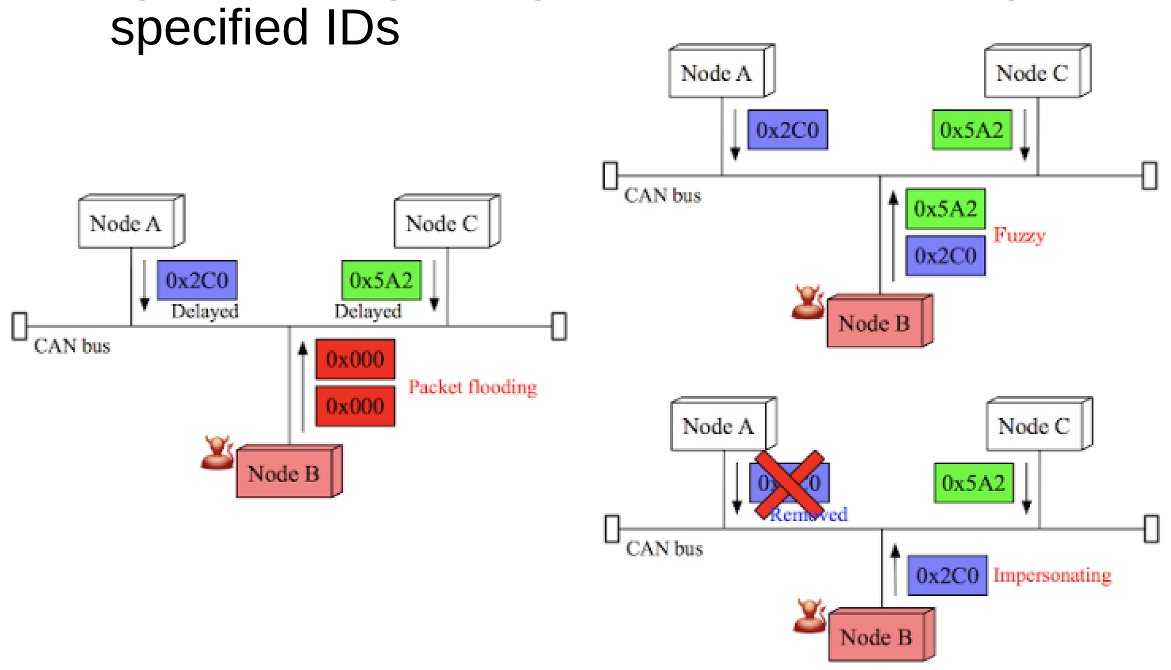
- Existing work [2]
  - Propose a two-stage anomaly detection system using a rule-based and ML model
  - Proves using Decision Tree, Random Forest, and XGBoost is an efficient method on the OTIDS dataset [3]

- Cybersecurity
  - Vehicles are becoming more connected to the internet and need methods for identifying malicious attacks / messages.

- Goals of project: simulate an anomaly detection system (ADS) in a vehicle environment
  - Combination of 2 filters:
    1. Machine Learning filter
       - Lightweight model
         - Hyper-parameter tuning
    2. Rule-based filter
  - Simulate ADS with Python-can

## Methodology

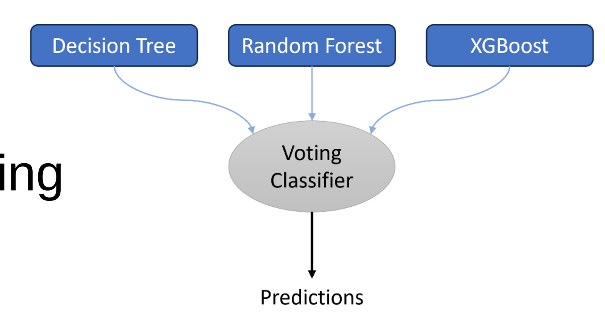
- Attack Types [3]
  - DoS: flooding with communication
  - Fuzzy: sniff network to create passable randomized CAN ID & DATA payload
  - Impersonation: malicious node stops messages by controlling a target node and inserting specified IDs




- Rule-based model
  - Implemented rule set:
    - Arbitration ID: Check for illegal ID
    - Message frequency: Compare message frequency for certain ID
    - Sequence: Compare sequence of IDs
- ML Model
  - OTIDS dataset
  - Feature extraction
  - Data preprocessing
  - Classifier model
    - Decision Tree (DT)
    - Random Forest (RF)
    - XGBoost (XGB)
  - Hyper-parameter tuning

| Attack Type | Number of Instances |
|---|---|
| Attack Free | 2,369,868 |
| DoS | 656,579 |
| Fuzzy | 591,990 |
| Impersonation | 995,472 |

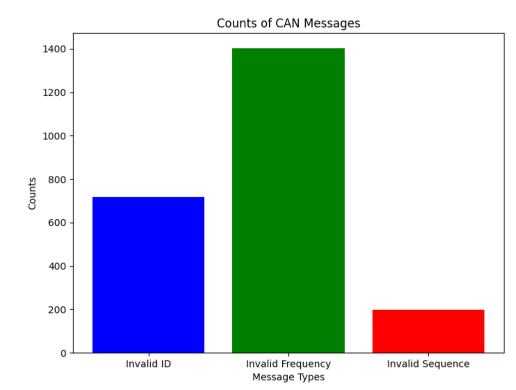
| Feature | Description |
|---|---|
| Timestamp | Recorded time |
| CAN ID | Identifier of CAN message in HEX |
| DLC | Number of data bytes |
| DATA[0~7] | Data value |

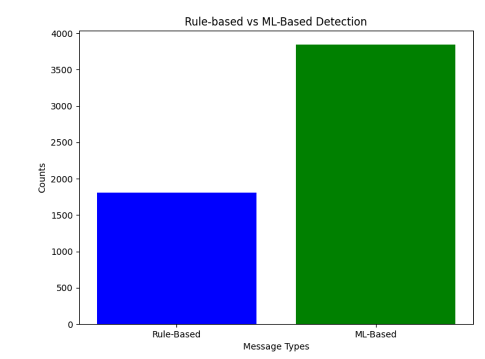



- CAN Network
  - Multi-ECU network (ECM, BCM, etc.)
  - Gateway ECU implements both Rule and ML filter
  - Random selection of 10,000 messages, comprised of both valid and invalid messages
  - Messages propagate in through gateway, valid messages pass through, invalid are marked

## Results

- CAN Network: Rule-based Filter
  - Simplistic rule filtering is ineffective



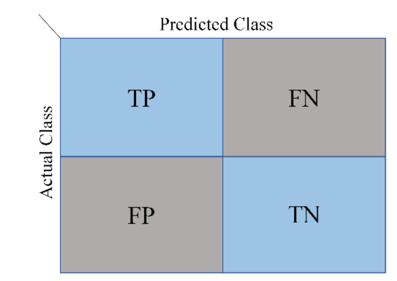




- CAN Network: ML-based filter

| Rule-Based % | ML-Based % |
|---|---|
| 0.480514096185738 | 0.802860696517413 |

- ML Model
  - Training Time
  - Confusion Matrix [4]

| Confusion Matrix | Attack | No Attack |
|---|---|---|
| Attack | 231,961 | 4,453 |
| No Attack | 11,927 | 213,003 |




  - Validation Equations [2]
    - Accuracy
    - Detection Rate
    - False Alarm Rate
    - F1 Score

$$Acc = \frac{TP+TN}{TP+TN+FP+FN}$$
$$DR = \frac{TP}{TP+FN}$$
$$FAR = \frac{FP}{TN+FP}$$
$$F1 = \frac{2*TP}{2*TP+FP+FN}$$

  - Validation Results

| No Attack | Result |
|---|---|
| Accuracy | 96.45% |
| Detection Rate | 95.11% |
| False Alarm Rate | 2.05% |
| F1 Score | 0.966 |
| **Attack** | **Result** |
| Accuracy | 96.45% |
| Detection Rate | 97.95% |
| False Alarm Rate | 4.89% |
| F1 Score | 0.963 |

  - Tuned Hyper-parameters
    - DT
      - criterion='entropy'
      - max_leaf_nodes=1000
      - min_samples_leaf=2
    - RF
      - n_estimators=30
      - max_leaf_nodes=1000
      - max_features=None
    - XGB
      - max_features=100
- Lightweight model
  - Size reduction from 698MB to 5.1MB
  - Pickle format

- Tuning Hyper-parameters for ML Model
  - Tuned by iterating through ranges for each parameter
  - Documentation for each ML method investigated to determine parameter
  - Example of a tuned parameter 'num_trees'

| XGBoost | | | | |
|---|---|---|---|---|
| num_trees | No Attack Accuracy | No Attack F1 Score | Attack Accuracy | Attack F1 Score |
| 20 | 96.555282% | 96.679981% | 96.555282% | 96.420850% |
| 30 | 96.615757% | 96.739868% | 96.615757% | 96.481823% |
| 50 | 96.619659% | 96.744388% | 96.619659% | 96.484992% |
| 100 | 96.657592% | 96.781855% | 96.657592% | 96.523347% |
| 125 | 96.653473% | 96.778380% | 96.653473% | 96.518490% |
| 150 | 96.661710% | 96.785934% | 96.661710% | 96.527498% |
| 175 | 96.658242% | 96.782743% | 96.658242% | 96.523718% |
| Optimal Value | 96.661710% | 96.785934% | 96.661710% | 96.527498% |

## Conclusions

- ADS performance in classifying an attack
  - ML filter performed better than rule based filter
    - During simulation the combined performance was 85% efficient in identifying an attack
- Future Work:
  - Work with ECU team at General Motors to develop calibration set to disable message authentication code (MAC) check at the ECU.
  - Evaluate ADS in vehicle using a NeoVi and Vehicle Spy to read and send messages into the vehicle CAN bus.
  - Implement more detailed rule-based schema to improve capture efficiency of rule-based filter.
  - Tune further to reduce ML model size and export in universal format

## References


1. P. G. A. G. Marco Di Natale, Haibo Zeng, Ed., Understanding and Using the Controller Area Network Communication Protocol. New York, NY: Springer, 2012
2. Purohit and M. Govindarasu, "Ml-based anomaly detection for intra-vehicular can-bus networks," in 2022 IEEE International Conference on Cyber Security and Resilience (CSR), 2022, pp. 233–238.
3. H. Lee, S. H. Jeong, and H. K. Kim, "Otids: A novel intrusion detection system for in-vehicle network by using remote frame," in 2017 15th Annual Conference on Privacy, Security and Trust (PST), 2017, pp. 57–5709
4. Y. Yalman, T. Uyanık, I. Atli, A. Tan, K. Bayindir, Karal, S. Golestan, and J. Guerrero, "Prediction of voltage sag relative location with data-driven algorithms in distribution grid," Energies, vol. 15, p. 6641, 09 2022