CAN Bus Anomaly Detection Using Machine Learning Models

Presenting by :

Saeed Amiri, Anas , Sravani Hukumathi Venkata

Introduction:

The **Controller Area Network** (CAN) bus is the standard communication protocol for reliable, efficient data exchange between Electronic Control Units (ECUs) in modern vehicles.

Developed in 1983, the CAN bus is exceedingly robust in physical transmission but exceedingly insecure in its design.

However, as vehicles become increasingly connected, the CAN bus's inherent lack of security controls has become a critical vulnerability.

This project focuses on developing an automotive Intrusion Detection System (IDS) to secure these networks against attacks.

Problem Statement and Challenges:

1. CAN Protocol has many security Vulnerabilities.

THREAT 1

No Authentication

Any device plugged into the bus (e.g., via OBD-II) can impersonate critical systems like Brakes or Engine. This is **Spoofing**.

THREAT 2

Priority Abuse (DoS)

An attacker can flood the bus with ID `0x000`. Since 0 is dominant, no other node can speak. The car is paralyzed.

THREAT 3

No Encryption

All data is plaintext. Attackers can easily "sniff" traffic to reverse-engineer proprietary signals.

Problem Statement and Challenges:

2. Dataset Challenges.

- A significant hurdle in developing robust Machine Learning (ML) based IDS is the shortage of adequate, publicly available datasets. ML models require substantial amounts of diverse training data to generalize well.
- This project addresses this challenge by utilizing the can-train-and-test dataset, which provides a comprehensive benchmark for evaluating IDS performance against specific attack vectors.

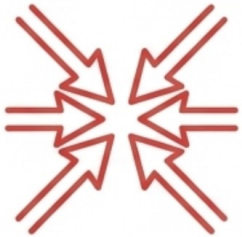
Our Goal: A Unified System to Detect Two Distinct Attack Signatures

The challenge was to create a single anomaly detection system capable of identifying two very different types of attacks with high precision, with/without relying on pre-existing attack labels.

DoS Attacks

Characterized by high frequency and dense traffic volume, designed to overwhelm the bus.

.



Fuzzy Attacks

Characterized by random payload injection, irregular timing, and spoofed (fake) message IDs.

CAN Protocol: The Overview

Controller Area Network (CAN) is the "nervous system" of modern vehicles, allowing ECUs (Electronic Control Units) to communicate without a central host.

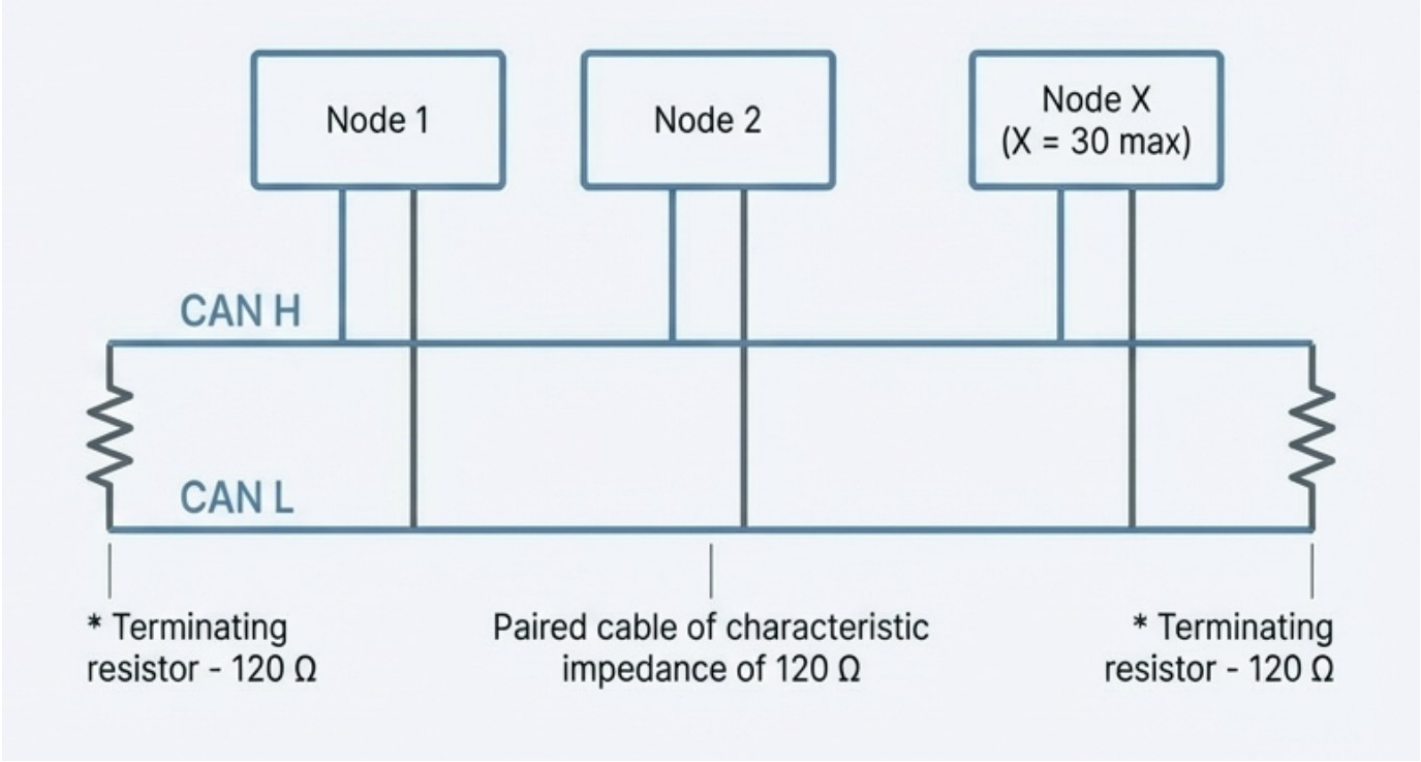
Broadcast Topology: All nodes share two twisted wires (`CAN_H`, `CAN_L`). Everyone hears everything.

Differential Signaling: Highly resistant to electrical noise in the engine bay.

No Addressing: Messages identify *content* (e.g., "Engine Speed"), not the sender.

Real-time: Designed for short, high-priority control messages.

CAN Bus Network:



CAN Protocol: Data Overview

We analyze specific fields to detect anomalies:

Arbitration ID (11-bit): Determines priority. Lower ID = Higher Priority (0x000 > 0x7FF).

DLC (4-bit): Data Length Code. How many bytes follow?

Data Field (0-8 Bytes): The actual sensor payload.

Our Strategy: We Transformed Raw Data into High-Signal Features

Machine learning models cannot interpret raw CAN bus data directly. The key to success was to engineer specific mathematical features that act as "fingerprints" for normal and malicious behavior. We created two distinct sets of features tailored to the unique signatures of DoS and Fuzzy attacks.

Raw data:

timestamp	can_id	data_field	attack
1.672531e+09	199	0FFF0FFFF00100FF	0
1.672531e+09	1E5	460196E001FE6701	0
1.672531e+09	0C9	0000000A00000000	0
1.672531e+09	0C9	8412BF0A00010000	0
1.672531e+09	191	061408F508F50000	0

Feature Engineering:

1. Merge all the train dataset files into a single file.
2. Transform the timestamp into IAT(Inter Arrival Time) for each frame of unique CAN ID.
3. Convert the CAN ID from hex format to decimal format.
4. Split 8 bytes of data into 8 columns and convert hex format of data into decimal format.
5. Set the attack column with numbering for different type of frames like Normal (0), DoS attack frames(1), Fuzzy attack frames(2).

Advanced Feature Engineering Strategy:

We transform raw hex logs into behavioral metrics:

For DoS Detection (Time):

IAT (Inter-Arrival Time): Time gap between messages.

Frequency (Hz): 1 / IAT. DoS attacks show massive spikes (>2000Hz).

For Fuzzy Detection (Payload):

Rolling Volatility: Standard deviation of data bytes (window=5). High volatility = Random injection.

Hamming Distance: Bit flips between frames.

Is New ID: Flag if ID was never seen in training.

Advanced Feature Engineering:

Fuzzy attacks are detected by analyzing the content and context of the message payloads.

Unknown ID Detection ('is_new_id")

• **Logic:** We build a "whitelist" of all valid CAN IDs from normal training data. Any message with with an ID not on this list is flagged.

• **Why it Works:** Attackers often invent random IDs. This is a powerful, deterministic of spoofing.

Rolling Volatility

• **Logic:** Calculate the standard deviation of payload bytes over a sliding window of 5 messages for the same ID.

• **Why it Works:** Normal sensor data changes smoothly (e.g., 50, 51, 52 -> Low Volatility). A fuzzy attack injects random noise (e.g., 255, 0, 12 - High Volatility).

Hamming Distance

• **Logic:** Measures how many bits have flipped between the current and previous message payloads for the same ID.

• **Why it Works:** In normal operation, Low Distance (2 bits) only a few bits change between updates (e.g., speed from 60 to 61 ->Low Distance). Random data injection causes many bits to flip (High Distance).

Advanced Feature Engineering:

Denial-of-Service attacks are defined by their timing and frequency, not their content.

Frequency (frequency_hz)

Logic: We calculate the inverse of the rolling mean
inter-arrival time (1 / IAT).

• **Why it Works: T**his is the "Smoking Gun" for DoS. Normal messages have a stable frequency (e.g., 10Hz), while a DoS attack creates a massive spike (e.g., >10,000 Hz). A small 'epsilon' value is added to the denominator to prevent division-by-zero errors.

Logarithmic IAT ("log_iat')

• **Logic:** Apply the natural logarithm to the Inter-Arrival Time (IAT).

• **Why it Works:** IAT values can have a very wide distribution. The log function "squashes" this range, making it easier for models to learn patterns
patterns without being skewed by extreme outliers.

Machine Learning Model Approaches:

To detect **Normal, DoS, Fuzzy, and Unknown CAN ID attacks** in automotive CAN networks using **Supervised, Unsupervised, and Hybrid Machine Learning models**.

- **Supervised & Semi-Supervised Approaches:**
  - **K-Nearest Neighbors (KNN) —** Pattern-based classification
  - **One-Class SVM —** Learns normal behavior, flags anomalies
  - **XGBoost —** High-accuracy ensemble classifier

- **Unsupervised & Rule-Based Approaches**
  - **Isolation Forest —** Rare anomaly detection
  - **LOF (Local Outlier Factor) —** Density-based outlier detection
  - **New CAN ID Detection —** Identifies spoofed or unseen ECU messages

1. Hybrid Ensemble of Unsupervised Models

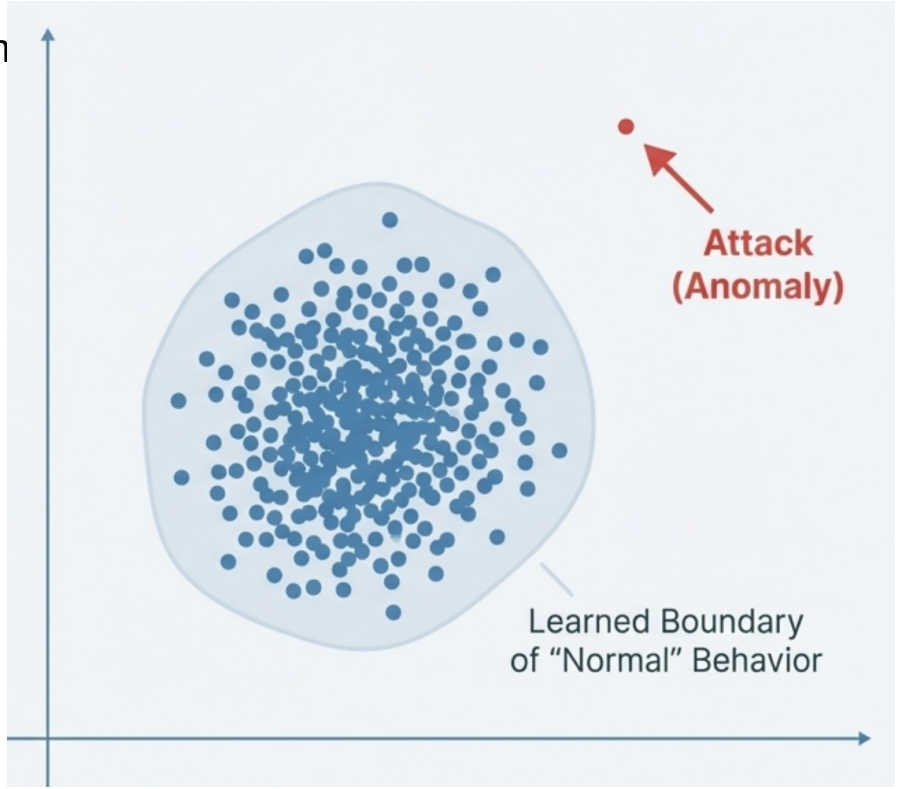
Our approach is a form of One-Class Classification. We train our models only on what "normal" traffic looks like.

The Strategy

1. The models learn the dense, complex boundary legitimate CAN bus behavior.
2. When an attack appears during testing, it falls outside this learned boundary and is flagged as an "anomaly."

Why this is effective

• In the real world, you don't know what a future attack will look like. By training only on normal data, the system is robust against no



Isolation Forest, the 'Global' Anomaly Detector

Role: Primary detector for large-scale anomalies like DoS attacks.

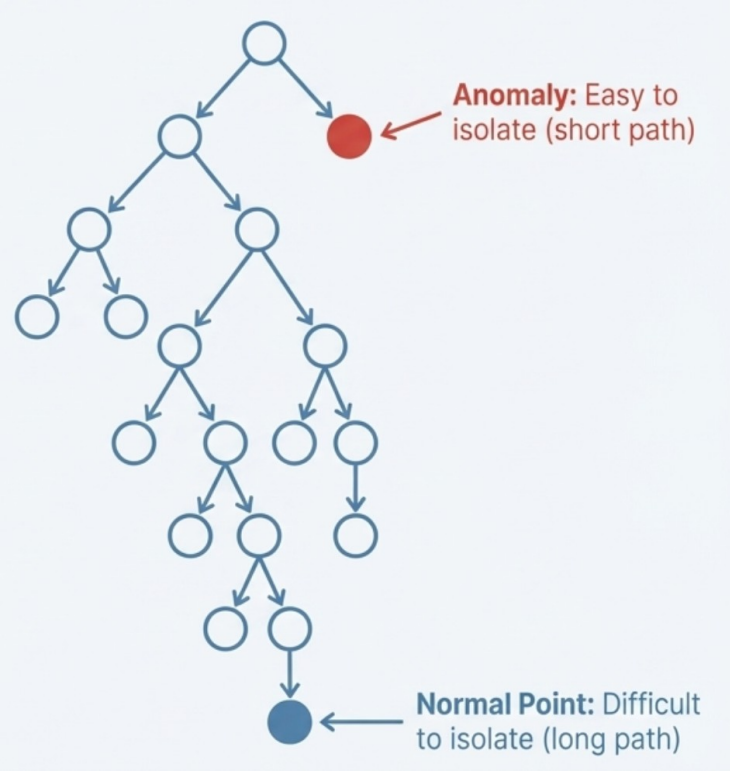
How it Works:

The algorithm builds a "forest" of decision trees. It works by explicitly isolating anomalies rather than profiling Normal data.

- **Normal Points:** Are deep within data clusters and require many random splits ("cuts") to be isolated. They have a long path in the tree.
- **Anomalies:** Are few and different, sitting at the edges of the data. They are easy to separate and require very few cuts to be isolated, resulting in a short path.

Training Strategy

Trained on the FULL normal dataset (~700k samples) to create a robust and precise boundary of normal behavior.



Local Outlier Factor, the 'Local' Anomaly Specialist

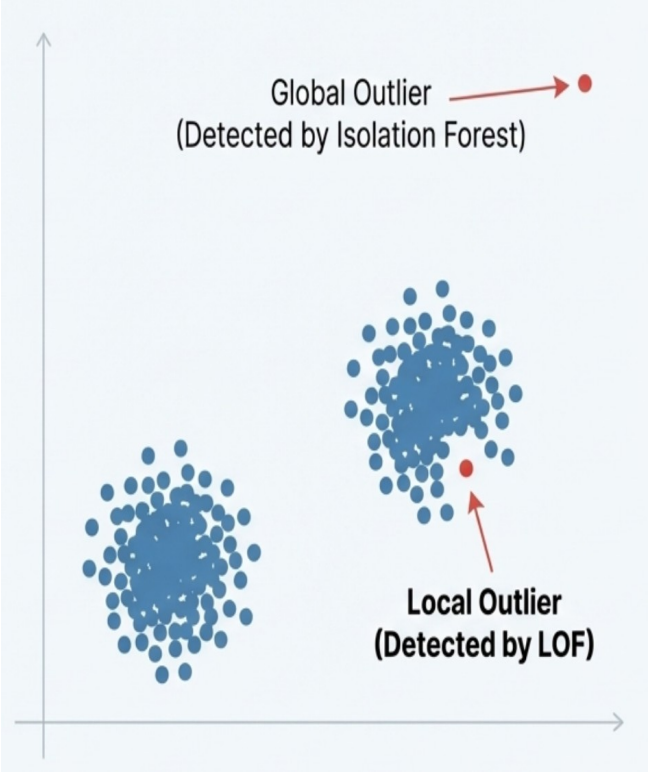
Role:
Specialist detector for subtle Fuzzy attacks that might otherwise look normal.

How it Works:
LOF is a density-based algorithm. It compares the local density of a data point to the density of its neighbors.

• **Anomalies:** Are points that have a substantially lower density than their neighbors, even if they aren't global outliers. It excels at finding points that are "isolated" within their own local cluster.

Training Strategy
Trained on a SUBSAMPLE (50k samples) of the normal data.

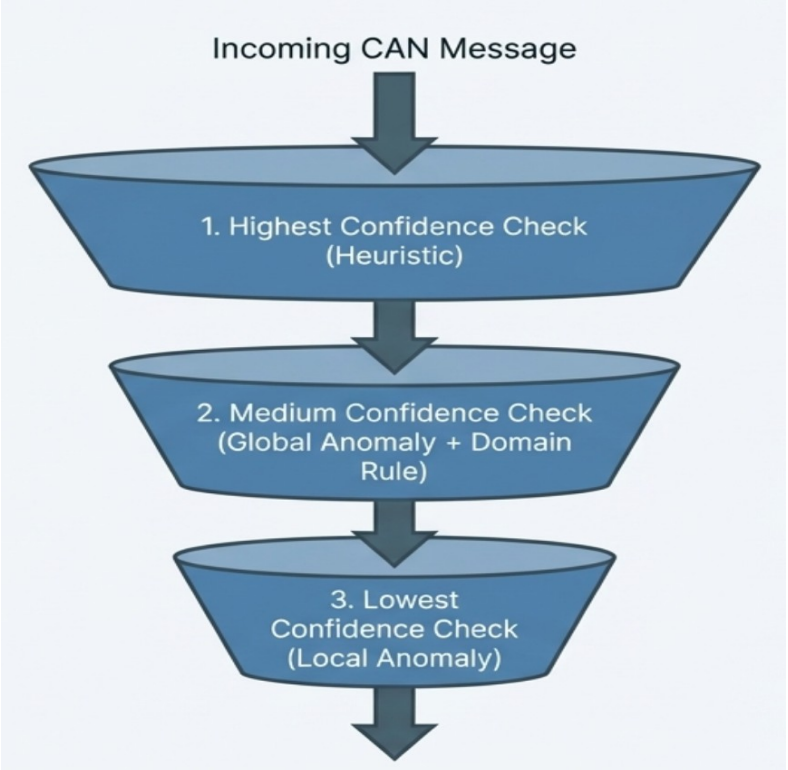
• Reason: LOF has high computational complexity (O(n2)). Training on a representative sample is a critical engineering decision to ensure feasible training times.



The Core Innovation: A Priority Hierarchy Logic for Classification

The most critical technique in our system is not the models themselves, but how their predictions are combined. Instead of treating models as equal voters, we process each message through a priority-based logic that moves from highest to lowest confidence checks.

The Benefit: This hierarchical approach prevents misclassification. For example, it stops a rapid-fire Fuzzy attack from being incorrectly labeled as a DoS attack simply because it has a high frequency.

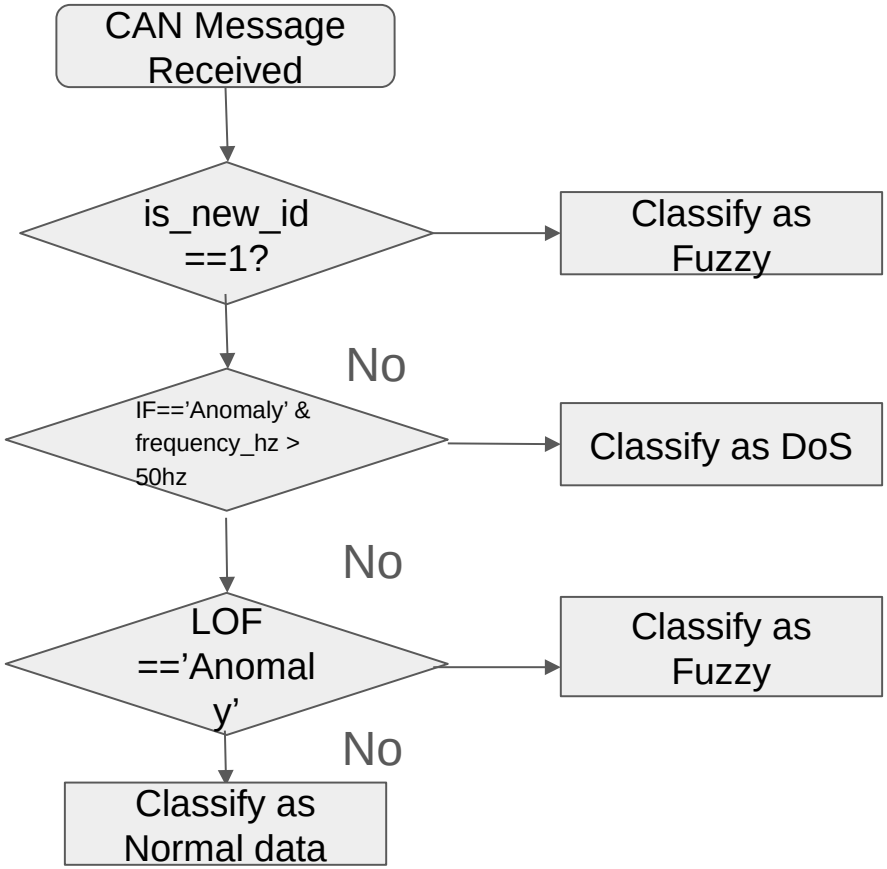


Unpacking Our Three-Tiered Detection Logic

Priority 1 - Heuristic Check

Priority 2 - DoS Check

Priority 3 - Statistical Check



The Result: A System with High Precision and Near-Zero False Alarms

This multi-layered, hybrid approach yielded exceptionally strong performance on the test dataset.

~99.96%

Normal F1 score:
We correctly identified almost all legitimate traffic, meaning virtually no false alarms for the user.

~99.20%

Fuzzy F1 score
The 'is_new_id' priority logic was key to capturing nearly all spoofing and random data injection attacks.

~98.00%

DoS F1 score
The system accurately distinguished true DoS floods from other anomaly types like high-frequency Fuzzy attacks.