Lightweight Deep Learning for Cyber-Resilient Heavy Vehicles: Efficient Signal Reconstruction on Embedded Systems

MAXWEL BAR-ON, INDRAKSHI RAY, JEREMY DAILY

COLORADO STATE UNIVERSITY

HOSSEIN SHIRAZI

SAN DIEGO STATE UNIVERSITY

Outline

- Problem Motivation
- Approach Overview
- Approach Details
- Conclusion

Problem Motivation

Heavy Vehicles Are Under Growing Cyber Threats

- Heavy trucks transport $13 trillion worth of goods annually in the U.S.
- Modern trucks rely heavily on embedded computers (ECUs)
- ECUs communicate over protocols which can be attacked
- Increasing use of wireless features (GPS, telematics) creates remote attack surfaces
- Cyberattack on a truck can:
  - Disrupt braking, steering, or engine performance
  - Trigger costly downtime or even accidents
  - Cause ripple effects across the supply chain

Existing Defenses Are Not Enough

- Intrusion detection systems (IDS) can spot unusual signals
  → But they **do not restore** corrupted signals during an attack
- Protocol redesigns have been proposed
  → But they require **industry-wide changes** (not practical)
- Deep learning has been used to reconstruct corrupted signals
  → But current models are:
  - Too large for in-vehicle hardware
  - Trained separately for each signal type
  - Hard to scale and deploy in real vehicles

Overview of Our Approach

A Lightweight, Unified AI Model

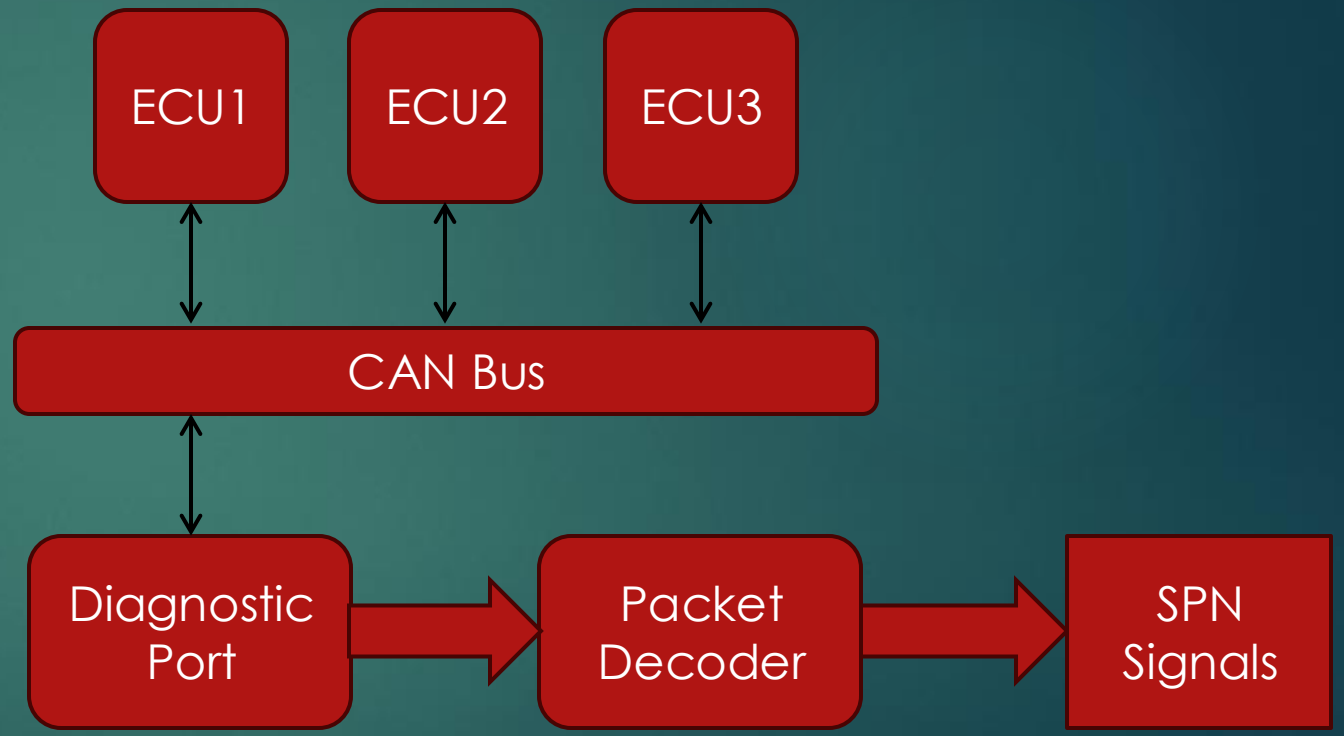
- A single compact model that reconstructs corrupted signals in real time
- Learns how normal vehicle behavior looks by watching many signals
- When under attack, fills in missing or fake data using trusted context
- Can run **directly on the vehicle's embedded hardware**
  - Uses **quantization** to shrink size and speed up computation
  - Less than **1MB** in size; handles many signals at once
- Works even under **realistic attack scenarios** affecting multiple components

What's New

- Trains one model for **all signals**, not one-per-signal
- Uses **random masking** during training for better generalization
- Applies **Quantization-Aware Training (QAT)** to reduce size by 4×
- Outperforms prior work on both **reconstruction accuracy** and **deployment efficiency**
- Validated on real truck data and realistic simulated attacks

Heavy Vehicle Network Architecture

- ECUs communicate over the CAN bus using J1939 packets
- Each packet contains multiple vehicle signals (SPNs)
- A packet decoder connects via the diagnostic port
- Decoder extracts SPN values using J1939 protocol rules



Generalizable Deep Learning for Embedded Vehicle Networks

- Train one model for all signals — no need for separate models per SPN
- Use **random feature masking** to simulate signal disruptions during training
  - Teaches the model to recover from diverse attack scenarios
- Model learns to infer missing SPN values using context from other signals
- Enables **real-time signal reconstruction** when signals are corrupted
- Improves generalization to **unseen attacks and multi-SPN disruptions**
- Avoids reliance on hardcoded correlations or signal-specific models

Efficient Enough to Run on Embedded Devices

- Apply **Quantization-Aware Training (QAT)** to reduce model overhead
- Represent model weights and activations with **8-bit integers** (vs. 32-bit floats)
  - 4× smaller model size
  - Up to **12.8× reduction in compute cost**
- Quantization simulated during training to retain accuracy
- Maintains **<1% reconstruction error** in most scenarios
- Final model is **under 1MB** — suitable for on-vehicle deployment

Details of Our Approach

Collected Data

- J1939 logs collected from a **Kenworth T270 Class 6** heavy vehicle
- 57-minute drive in **Colorado**
- Includes messages from **28 SPNs** across **18 PGNs**
- Converted to **multivariate time series**
- Sampled every **60 milliseconds**
- Using a **sliding window** approach for model input

SPN	Description	Min	Max	Unit
91	Accelerator Pedal Position #1	0	100	%
92	Engine Percent Load At Current Speed	0	250	%
5398	Estimated Pumping – Percent Torque	-125	125	%

Dataset Size

Window Size	Data Subset			Total
	Train	Validate	Test	
10	37457	5351	10702	53510
20	37380	5340	10680	53400
30	37303	5329	10658	53290
40	37226	5318	10636	53180
50	37149	5307	10614	53070
60	37072	5296	10592	52960

Signal Reconstruction Setup

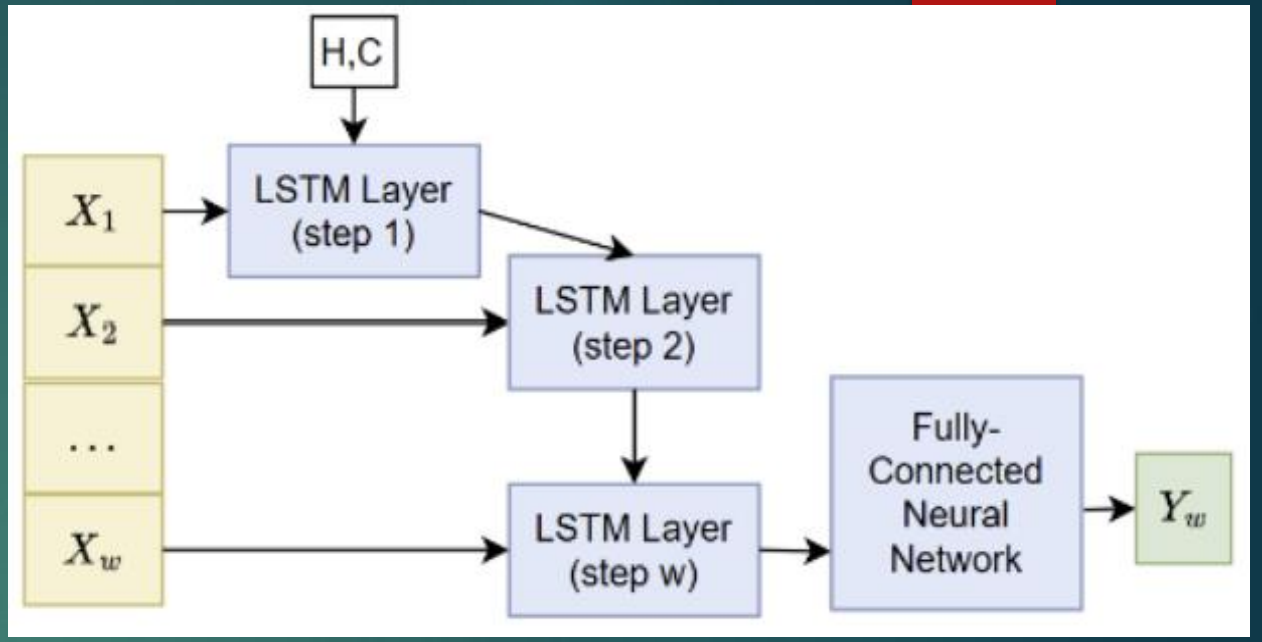
Model Input and Disruption Simulation

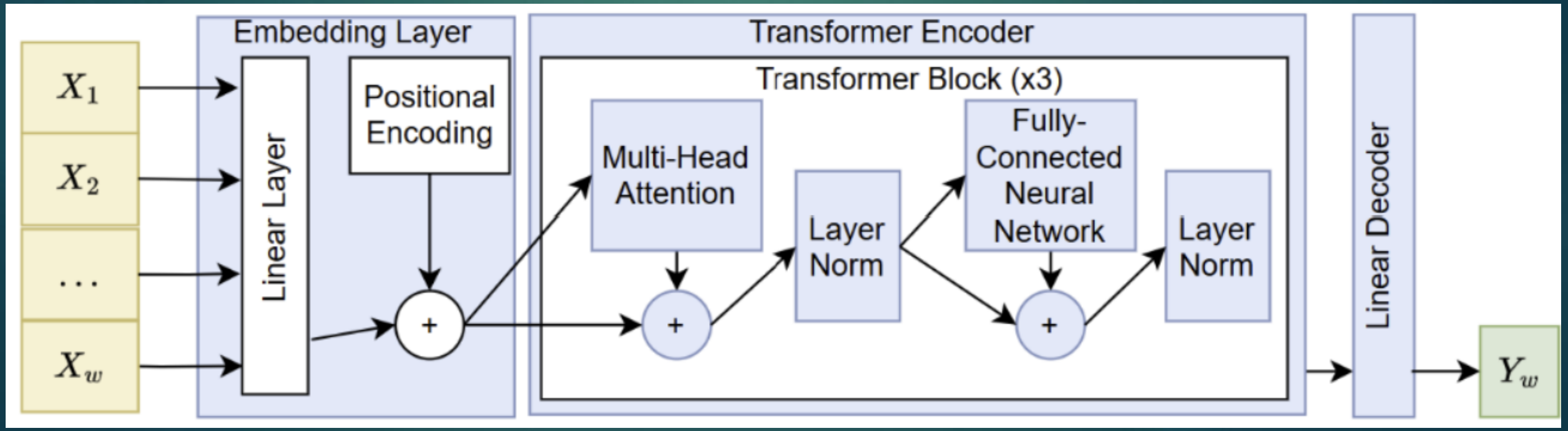
- Model input: **28 SPNs** tracked as a multivariate time series
- Each input sample: a **sliding window** of SPN values (e.g., last 10–60 timesteps)
- At each timestep, **randomly mask 10% of SPNs** by setting their value to **−1**
- Mimics real-world attacks where signals are dropped, corrupted, or suppressed
- Ensures model sees diverse disruption patterns during training

How the Model Learns to Recover Corrupted Signals

- Two model types evaluated: **Transformer** and **LSTM**
- Both take a window of recent SPN values as input
- Use unmasked signals to predict **all 28 SPNs at the final timestep**
- For masked SPNs, model output serves as the **reconstructed value**
- Output is a full 28-dimensional vector – supports **multi-signal recovery**
- Trained to minimize reconstruction error only on the **disrupted (masked)** SPNs

Architecture





Quantization-Aware Training (QAT)

- *Simulating Low-Precision Inference During Training*
- Simulates **8-bit precision** during training (weights & activations)
- Actual training uses **32-bit floats** for stability
- Quantization simulated using **clamping + rounding** to [−127, 127]
- Uses **scaling parameters** (1 for weights, 1 for activations per layer)
- Scaling learned from observed min/max values
- After training:
  - Weights stored as 8-bit integers
  - Activations quantized on-the-fly using learned scales

Robustness to Signal Disruption

- Goal: Evaluate model performance under partial signal loss
- Randomly mask SPNs in each time window (set to −1)
- Mask rates: 5% to 30% of SPNs per sample
- Model must reconstruct missing SPNs using context
- **100 evaluation trials**
- Models tested:
  - **Transformer** and **LSTM**
  - In both **Full Precision (FP)** and **Quantized (Q)** formats

Effect of Window-size

	Window Size	Transformer FP	Transformer Q	LSTM FP	LSTM Q
Base Mask Rate	10	0.0035	0.0036	0.0032	0.0043
	20	0.0027	0.0029	0.0035	0.0033
	30	0.0028	0.0029	0.0031	0.004
	40	0.0028	0.0029	0.0036	0.0058
	50	0.0026	0.0027	0.0055	0.0051
	60	0.0026	0.0027	0.0046	0.0066
Overall	10	0.0057	0.0058	0.0061	0.0062
	20	0.0042	0.0043	0.0082	0.0054
	30	0.0049	0.005	0.0117	0.0127
	40	0.0047	0.0049	0.0123	0.0192
	50	0.0045	0.0047	0.0156	0.0174
	60	0.0046	0.0047	0.0179	0.0179

Reconstruction error (MSE) of Transformer and LSTM models at different window sizes.

FP = full-precision (32-bit), **Q** = quantized (8-bit via QAT).

LSTM performs poorly with large window-sizes

.

Robustness to Signal Disruption

Mask Rate	Transformer FP	Transformer Q	LSTM FP	LSTM Q
10%	0.0028	0.0029	0.0039	0.0049
20%	0.0041	0.0043	0.0076	0.0089
30%	0.0096	0.0097	0.0349	0.0367

Reconstruction error (MSE) of Transformer and LSTM models at different mask rates.
FP = full-precision (32-bit), **Q** = quantized (8-bit via QAT).
Transformer shows lower error and stronger robustness across settings.

Transformer Outperforms LSTM, Even When Quantized

- **Transformer outperforms LSTM** across all mask rates
  - Especially strong at higher disruption levels (e.g., 30%)
- **Quantized models** maintain accuracy close to full-precision
- Transformer retains **<1% error** up to 30% disruption
- LSTM performance drops significantly under high mask rates
- QAT enables **efficient deployment** with minimal accuracy loss

Efficiency Gains through Quantization

		Transformer			LSTM		
		FP	Q	Savings	FP	Q	Savings
Computation Cost	w=10	15,812,320	2,721,700	5.81×	22,695,060	1,769,938	12.82×
	w=20	34,267,920	8,086,680	4.24×	45,250,260	3,526,658	12.83×
	w=30	55,387,520	16,115,660	3.44×	67,805,460	5,283,378	12.83×
	w=40	79,171,120	26,808,640	2.95×	90,360,660	7,040,098	12.84×
	w=50	105,618,720	40,165,620	2.63×	112,915,860	8,796,818	12.84×
	w=60	134,730,320	56,186,600	2.4×	135,471,060	10,553,538	12.84×
Storage Cost (KB)		2,690.43	820.96	3.28×	4,123.78	1,055.81	3.9×

Efficiency Gains Through Quantization

- Computation cost estimated using operation counts:
  - Count of 32-bit FLOPs vs. 8-bit integer multiplications
  - 32-bit floating-point ops consume ~18.5× more energy than 8-bit ops
  - Cost scales with window size and model complexity
- Quantization-Aware Training (QAT) improves efficiency:
  - Replaces many 32-bit operations with 8-bit integer multiplications
  - Scaling still requires a few 32-bit divisions per layer
- LSTM benefits more from quantization in compute savings:
  - Up to 12.8× lower compute cost due to more parameter-heavy design
- Transformer savings are moderate but consistent:
  - Compute savings range from 2.4× to 5.8×
  - Efficiency decreases with large windows due to attention's quadratic scaling
- Storage cost significantly reduced:
  - Transformer: from ~2.7 MB to ~820 KB (3.3× smaller)
  - LSTM: from ~4.1 MB to ~1.05 MB (3.9× smaller)

Our Method Outperforms State-of-the-Art Approach

- Prior work by Shirazi et al. [40]:
  - Trained **separate LSTM models per signal**
  - Relied on **manual feature selection**
  - High **storage cost**, not scalable
- Our approach:
  - Uses a **single unified model**
  - Trained via **random feature masking**
  - Supports **multi-signal reconstruction**
  - Works even in **quantized (8-bit) form**
- **Quantized Transformer outperforms** Shirazi et al. on all disruption levels
- Achieves up to **10× lower MSE** with **much lower model size**

Disruption Rate	Shirazi et al. MSE	Error %	Our Method (QT)– MSE	Error %
1 / 6	0.0293	2.93%	0.0029	0.29%
1 / 11	0.0254	2.54%	0.0029	0.29%
1 / 16	0.0246	2.46%	0.0036	0.36%

Attack Simulation

Attack Simulation Setup

- Simulated attacks target **entire PGNs** by masking affected SPNs during evaluation
- Each attack assumes **prolonged PGN disruption** (no recent values available)
- Attacker model:
- Gains access via **physical port or compromised wireless ECU**
- Launches **DoS or spoofing attack** on a critical PGN
- Goal: evaluate if the **model can reconstruct missing SPNs** using surrounding context from unaffected signals

Simulated Attack Scenarios

PGN	Attack Scenario	Disrupted SPNs	Attack Method	Recovery Method
61444	Engine speed signal disruption (EEC2 compromise)	190, 512, 513, 2432	DoS via PGN suppression	ML-based reconstruction of all SPNs
64908	DPF outlet pressure tampering (ACM compromise)	3610	Spoofing forged pressure values	ML-based reconstruction of SPN 3610
65266	Exhaust gas temp. manipulation (firmware attack)	183, 184	Fake low-temp values during load	ML-based reconstruction of SPNs 183, 184

Results – Simulated Attack Recovery

- **Quantized Transformer** reliably reconstructs disrupted signals across all three attacks
- **Single-SPN attack (64908)** yields the best performance (under 1% error)
- **Multi-SPN attacks (61444, 65266)** are harder to recover from due to reduced context
- All reconstructed signals remain within **acceptable error margins** for vehicle operation

Attack (PGN)	SPNs Affected	Avg. MSE	Avg. Error %
61444	190, 512, 513, 2432	0.0398	3.98%
64908	3610	0.0091	0.91%
65266	183, 184	0.0630	6.30%

Conclusion: Lightweight, Resilient, Real-Time Recovery

- Trained a **unified deep learning model** using random masking
- Applied **Quantization-Aware Training** for 4× smaller models, 12.8× compute savings
- Compared **Transformer vs. LSTM**:
  - Transformer: better accuracy, robustness, and lower storage
  - LSTM: better compute efficiency at larger windows
- Simulated **3 realistic cyberattacks**
  → Quantized Transformer reliably reconstructed disrupted signals

Future work

- Evaluate runtime efficiency on hardware-in-the-loop (HIL) platforms
- Simulate attacks on heavy vehicle test beds
- Explore alternative deep time series learning architectures to further reduce reconstruction error
- Evaluate other efficient deep learning techniques such as pruning and binarization