AI FRAMEWORK TO AUTOMATE REVERSE ENGINEERING OF THE SERIAL INTERFACE OF EMBEDDED ELEVATOR SYSTEM

STUDENTS: KUHU NAIN, AAKASH NAMBOODIRI, STEVEN GONG, BELINA WANG, AMANDA ZHANG

OTIS

Objective

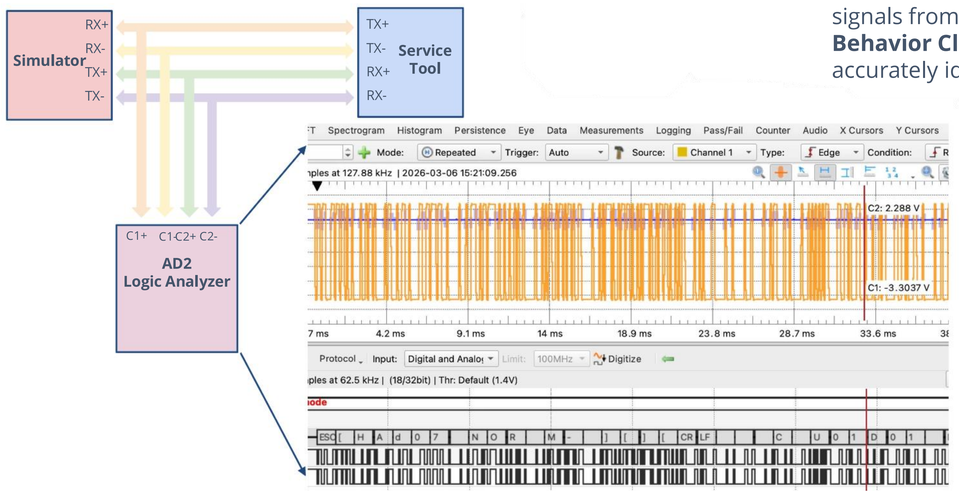
Background Elevator manufacturers use **proprietary protocols**, preventing OTIS technicians from efficiently maintaining competitor elevator system.
Solution: An AI-driven framework that leverages machine learning to **automatically decode unknown serial signals**, bypassing the need for manufacturer documentation.
Deliverables: A **classification system** for real-time elevator states and an interactive web interface that **maps the diagnostic navigation tree to signal sources.**

Primary Challenges

No Labeled Data: Relying on manufacturer-specific assumptions for manual labeling breaks generalization; the framework must remain **"blind" to function across different manufacturers**.
Protocol Structural Obscurity: Proprietary page boundaries and delimiters mean the system must **autonomously derive serial protocol logic** from raw bitstreams.
Dynamic Page Content: Rapid **data fluctuation**s during elevator movement complicate the **reliable identification and grouping of signals** belonging to the same page.

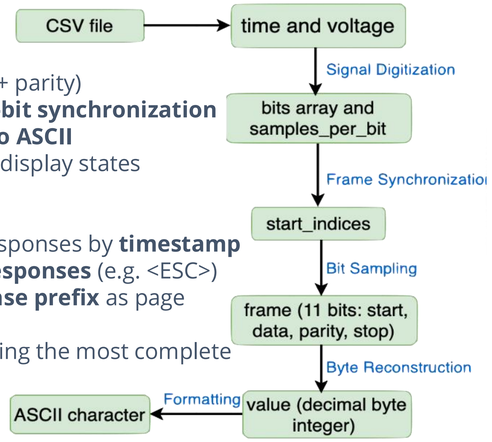
Data Collection

Signals are captured between the **E-411 elevator simulator** and the **OTIS Service Tool** using a **Digilent AD2 logic analyzer** as a passive tap.



Baud rate (~9700 bps) was determined by measuring the duration of a single bit (T ≈ 103µs), with odd parity for correct UART frame decoding.

Signal Segmentation & Preprocessing



Frame Analysis
- Identifies **UART** structure (start, data, stop bits + parity)
- Segments raw binary CSV into frames via **start-bit synchronization**
- Reconstructs frames into **bytes and decodes to ASCII**
- Output: 49-character **ASCII vectors** of elevator display states

Page Segmentation
- Pairs Service Tool keypresses with Simulator responses by **timestamp**
- Flags **structural tokens present in >50% of responses** (e.g. <ESC>)
- Uses majority-vote to find the **common response prefix** as page delimiter
- Merges responses with **>93% similarity**, retaining the most complete version

Signal & Labeling Pipeline Analysis

Data Preparation: Transformed raw CSV timing data into standardized 49-character ASCII vectors for consistent analysis.
Behavioral Categorization: Grouped captured signals into functional folders, including "Door Open," "Hall Call," and "Car Call."
Correlation Mapping: Applied agglomerative clustering to identify and group character indices that fluctuate simultaneously.
Refined Isolation: Employed a **two-pass clustering strategy** to effectively separate door-related signals from movement-related data.
Behavior Classification: Integrated changing-signal analysis with stable-state "fingerprints" to accurately identify simulator modes and operations.

Group Scoring Matrix

```
===== SELECTIVITY SCORE MATRIX =====
(Score = how much more active this group is here vs its average)
```

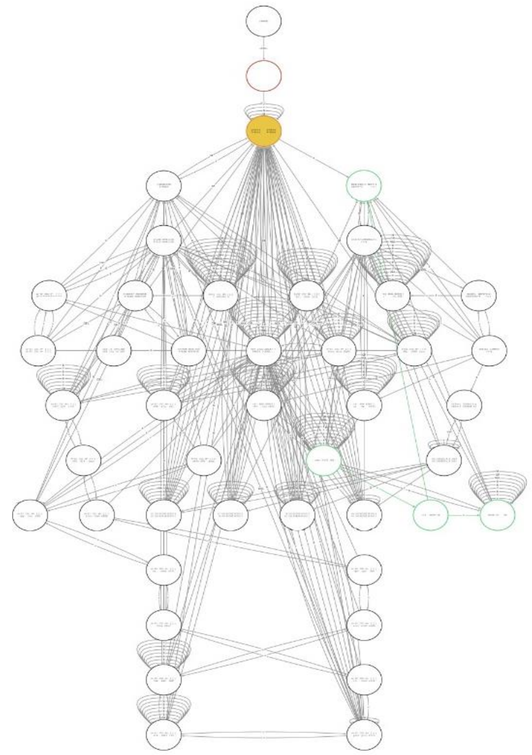
	CarCall	DoorOpenButton	DownHallCall	EmergencyButton	Randomize	UpHallCall
G0 [np.int64(2), np.int64(3)]	0.000	0.000	0.000	0.000	0.000	0.000
G1 [np.int64(5), np.int64(6), np.int64(7)]	-0.152	0.136	-0.165	0.518	-0.127	-0.210
G2 [np.int64(12), np.int64(13)]	0.018	0.230	0.075	-0.306	-0.067	0.051
G3 [np.int64(1)]	-0.076	0.333	0.249	0.062	-0.200	-0.369
G4 [np.int64(10)]	0.009	-0.280	-0.039	0.154	0.120	0.036
G5 [np.int64(25), np.int64(26)]	0.036	-0.056	0.018	0.033	-0.056	0.025
G6 [np.int64(27)]	-0.486	-0.486	0.514	-0.486	0.514	0.430
G7 [np.int64(31)]	0.106	-0.353	-0.186	-0.082	0.475	0.041
G8 [np.int64(34)]	0.134	-0.339	0.076	-0.339	0.544	-0.075

Navigation Tree Mapping

Automation Menu Mapping
- Constructs the elevator simulator screen states from RS232/RS422 serial communication
- B**uilds a hierarchical graph through depth-first exploration.**

FrameBuffer-based screen reconstruction
- Processes **partial terminal updates, cursor movements, and incremental redraws**
- **G**enerates stable UI states for tree analysis.

Adaptive exploration system
- **Captures, classifies, and replays serial interactions**
- Discovers menu transitions and hidden interface states **without prior protocol knowledge.**



Interactive Web Interface

- Connects to backend systems, essentially replacing service tool
- Initiate menu tree construction
- Trace path via graph to signals' pages in navigation tree

Future Work

- Test the framework for **non-OTIS service controllers**
- Improve **accuracy of signal labelling**
- Expand web interface for technicians/users to **incorporate labeling**

ELECTRICAL & COMPUTER ENGINEERING
UNIVERSITY of WASHINGTON

ADVISERS: PROF. MAHMOOD HAMEED
SPONSOR: OTIS ELEVATOR COMPANY
OTHER ACKNOWLEDGEMENTS: PROF. JENQ-NENG HWANG, ZACH HAO