



Safeguarding Embedded Controllers through Side Channel Analysis

US

aps SRP
ASU
Nexant
ARIZONA ISRAEL TECHNOLOGY ALLIANCE

Arizona

Colorado • NREL

Georgia • Georgia Tech

Tennessee • Deltek

Massachusetts • GE • Fova Energy

Michigan • OPAL-RT TECHNOLOGIES

Pennsylvania • DLC

Washington • SEL

Washington, DC • MITRE



Israel

Ben-Gurion University of the Negev

OTORIO

ARAVA POWER

CONTEL TECHNOLOGIES

RAD

SIGA OT Solutions

MRC ALON IVOR POWER

• Core Team Member
• Team Member (Volunteering Work)
• Advisory Board

Task 13

- PI: Prof Yossi Oren
- Michael Amar

3/12/24

Task Goals

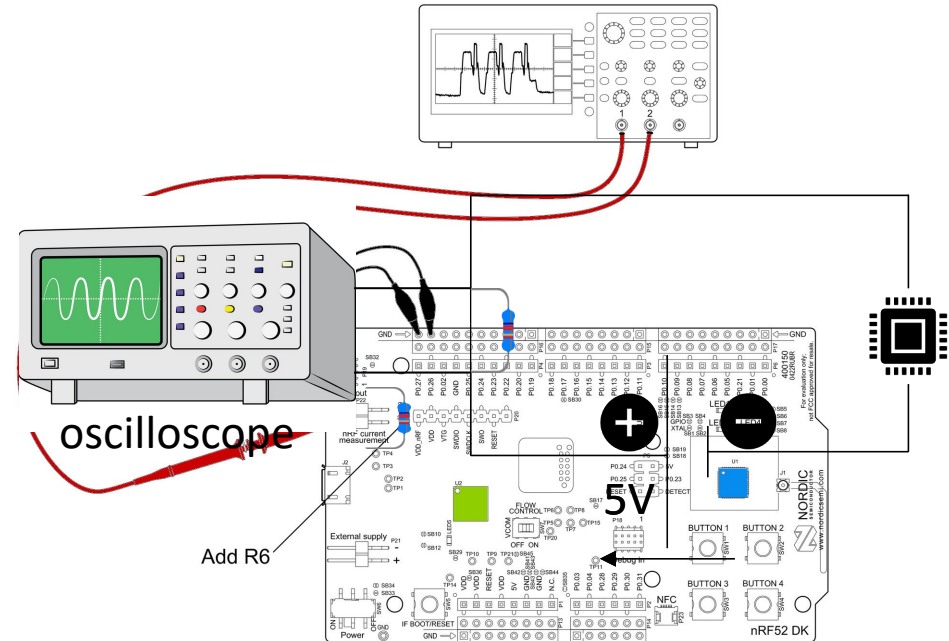


- Firmware Verification on edge devices (programmable controllers)
- Monitor code execution
- Maintain low overhead

The power side channel



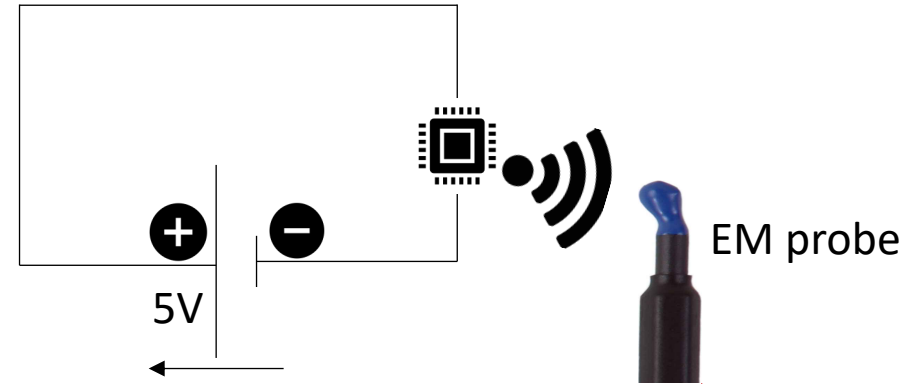
- Executing instructions causes transistors to switch on and off
- Ohms Law: $V = R * I$
 - Voltage is constant
 - Transistors switching causes varying resistance \rightarrow varying current
- The transitions cause fluctuating power consumption
- Different **instructions** consume power differently



The Electromagnetic side channel

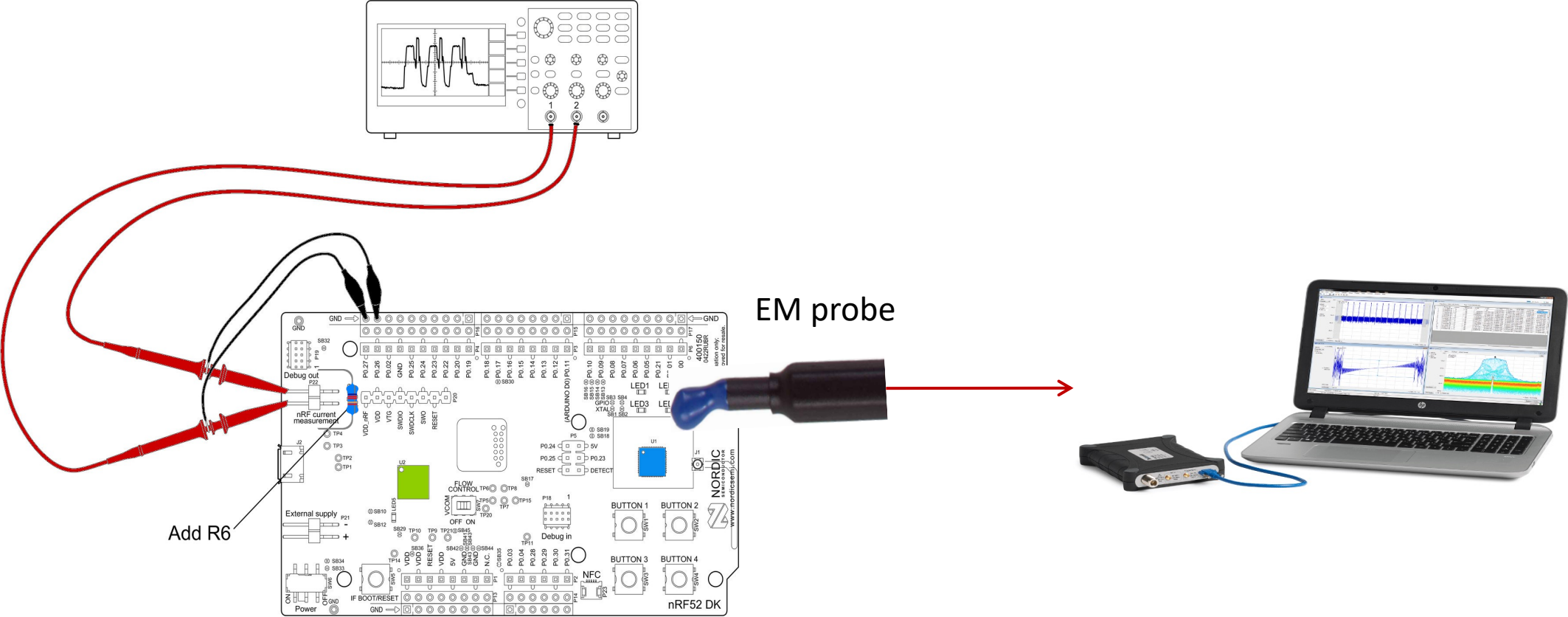
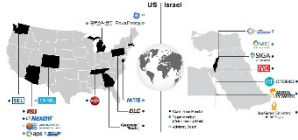


- Transistors switching cause varying resistance
→ varying current
- Any metallic substance becomes an antenna
- Current variations are translated into EM waves
- Wave characteristics depends on the power consumption (which depends on the executed instructions)



Spectrum Analyzer

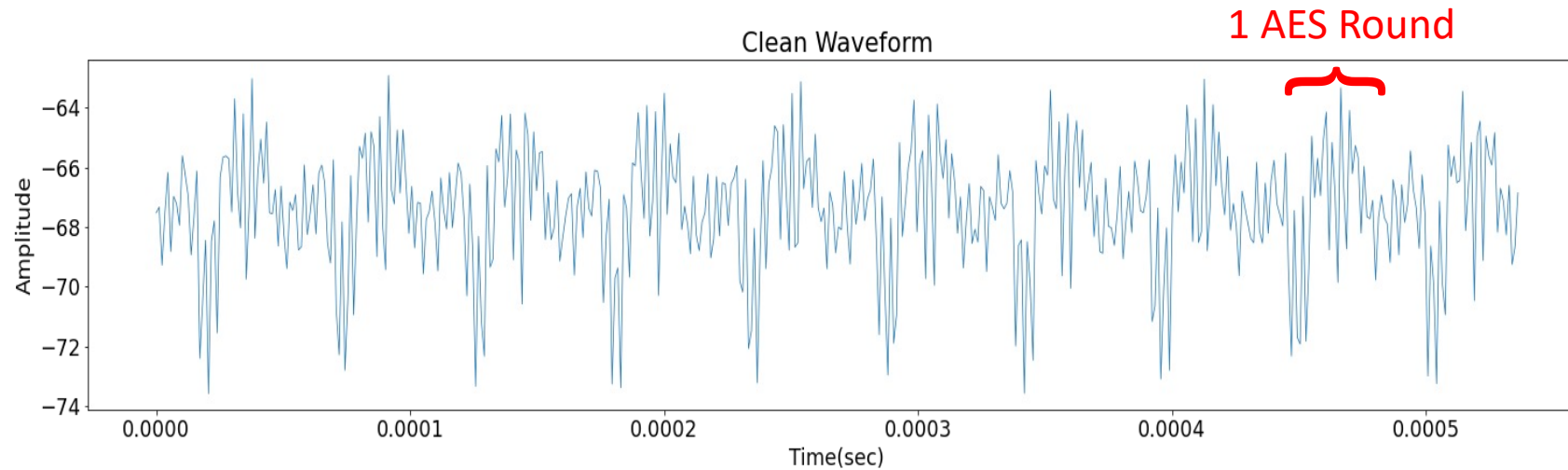
Experimental Environment



Example



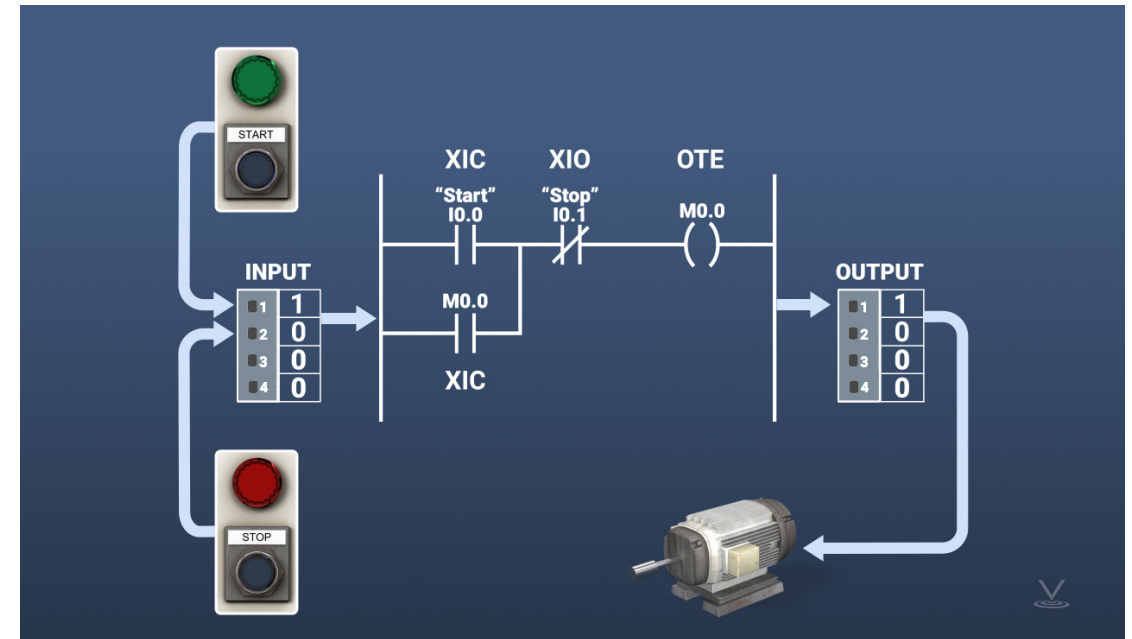
- EM signal
- Taken while executing AES encryption
- Reduced noise with FFT filter



Task 13 Goal



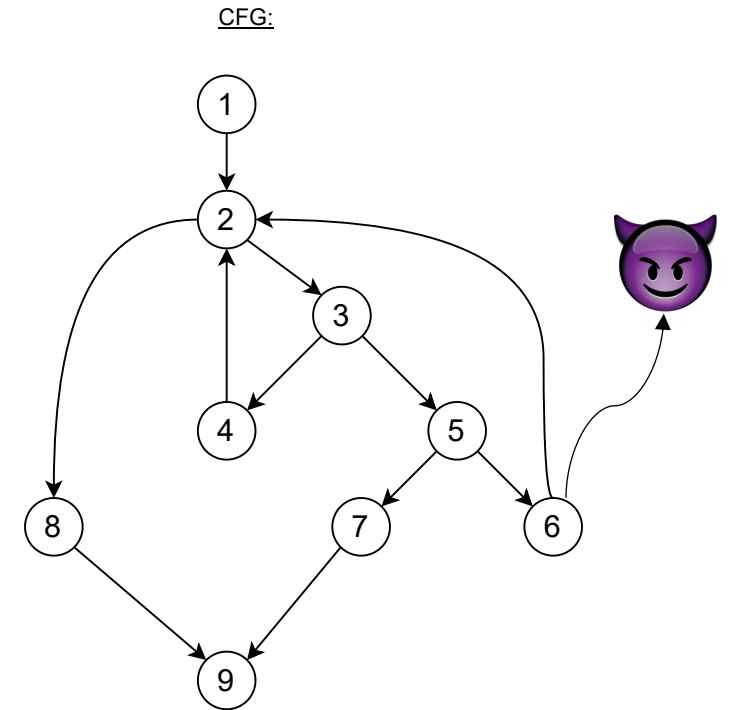
- Detect code hijackings
- Focus on Programmable Logic Controllers
 - Used to automate industrial processes
 - Used in power stations, water facilities, oil facility ...
 - Programmed with Ladder Logic
 - Code is rarely updated



Creating a behavioral baseline



- One program have multiple flows
- We want to characterize all
- Represent code as a control flow graph (CFG)
 - Node = Basic blocks
 - Directed edge = transition between blocks
- Control Flow Integrity
 - Verify transitions are legitimate
 - Common approach is instrumentation



Generating Test Cases



- Use static analysis tools
 - Symbolic execution engine

- Example:
To follow $1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 7 \rightarrow 9$, we need:
 $\{low \leq high, \quad x = v[mid]\}$

- Satisfiability solvers (SAT) returns actual assignment to the variables

- Repeat this process for all control flows*

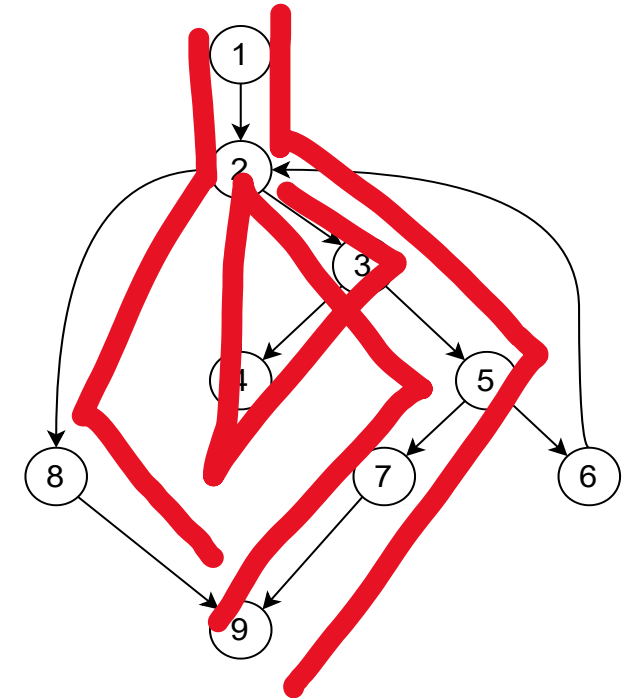
🚩 Number of flows explodes very fast

🚩 SAT problems are NP-complete

Source Program:

```
int binsearch(int x, int v[], int n)
{
  int low, high, mid;
  low = 0;
  high = n - 1;
  while (low <= high) | 2
  {
    mid = (low + high) / 2;
    if (x < v[mid])
      high = mid - 1; | 4
    else if (x > v[mid])
      low = mid + 1; | 6
    else return mid;
  }
  return -1; | 8
} | 9
```

CFG:



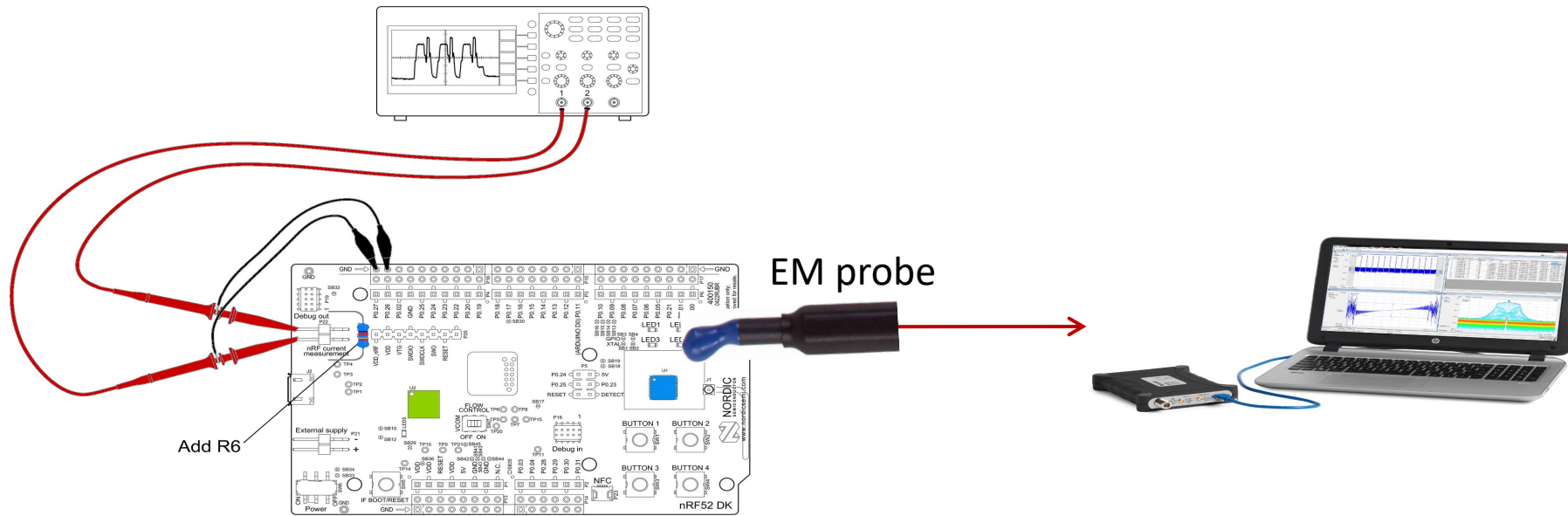
* program and test cases were taken from:

[Yi Han et al. **Hiding in Plain Sight? On the Efficacy of Power Side Channel-Based Control Flow Monitoring**]

Creating a dataset



- Each test case (control flow) was executed multiple times
- Collect EM & power signals simultaneously

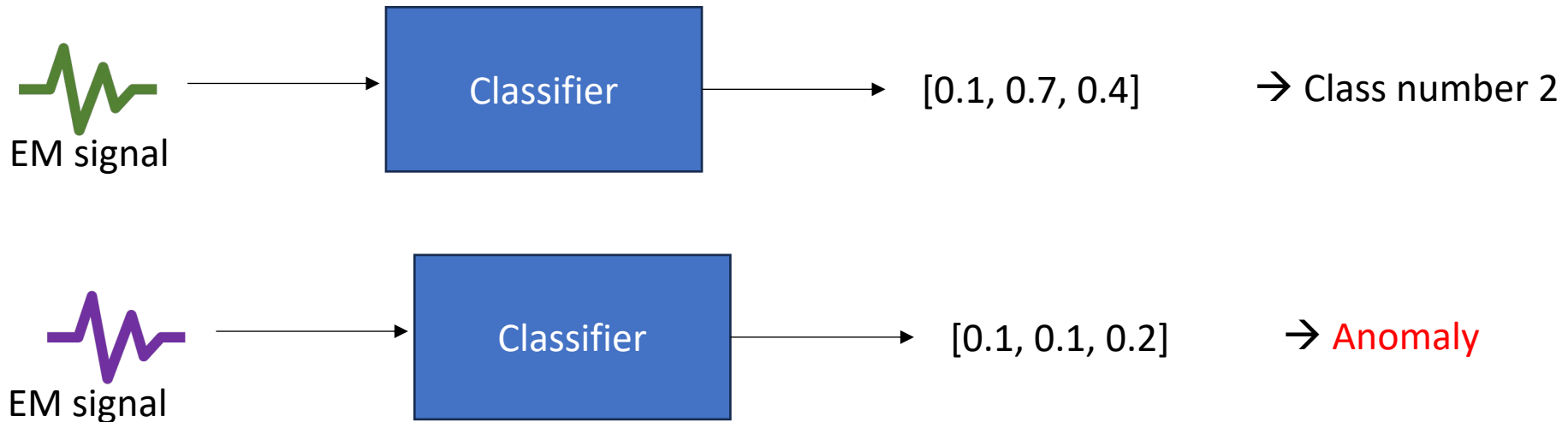


Anomaly detection



- Problem is translated into a classification problem
 - Each control flow is a class
 - Anomaly is low confidence in all classes
 - No need for anomaly samples!

- Examples:

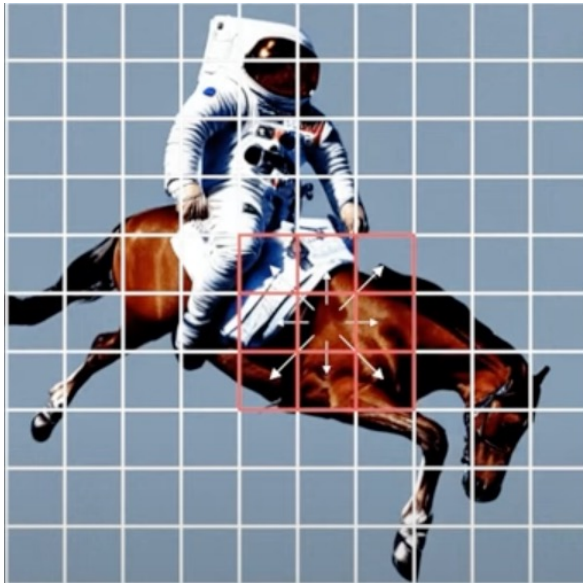


Classifier architecture



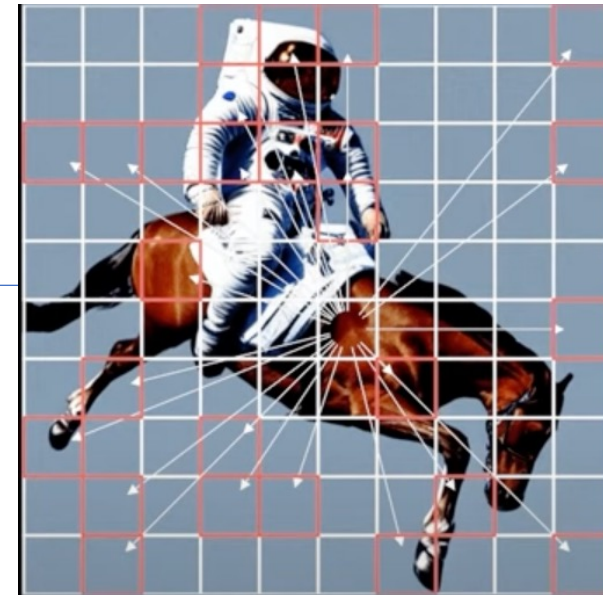
- Transformer based classifier
- Based on self attention mechanism
- Revolutionized the NLP world
- Faster than traditional RNNs

CNN



CNN edge detection Filter

Attention

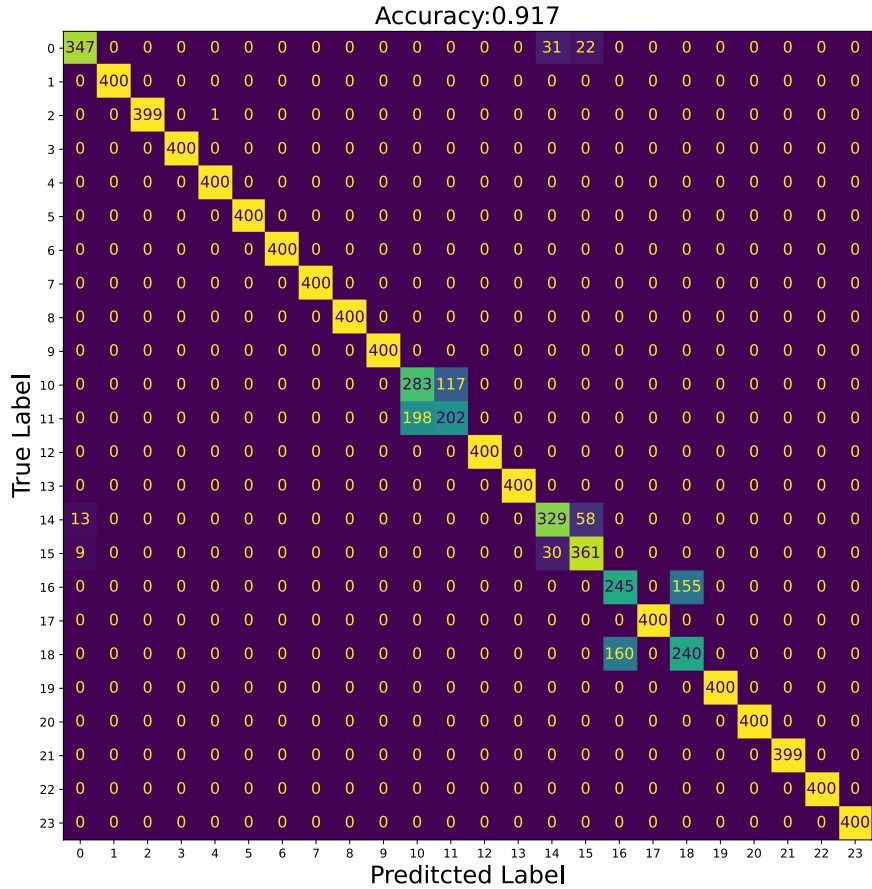


Our Dataset

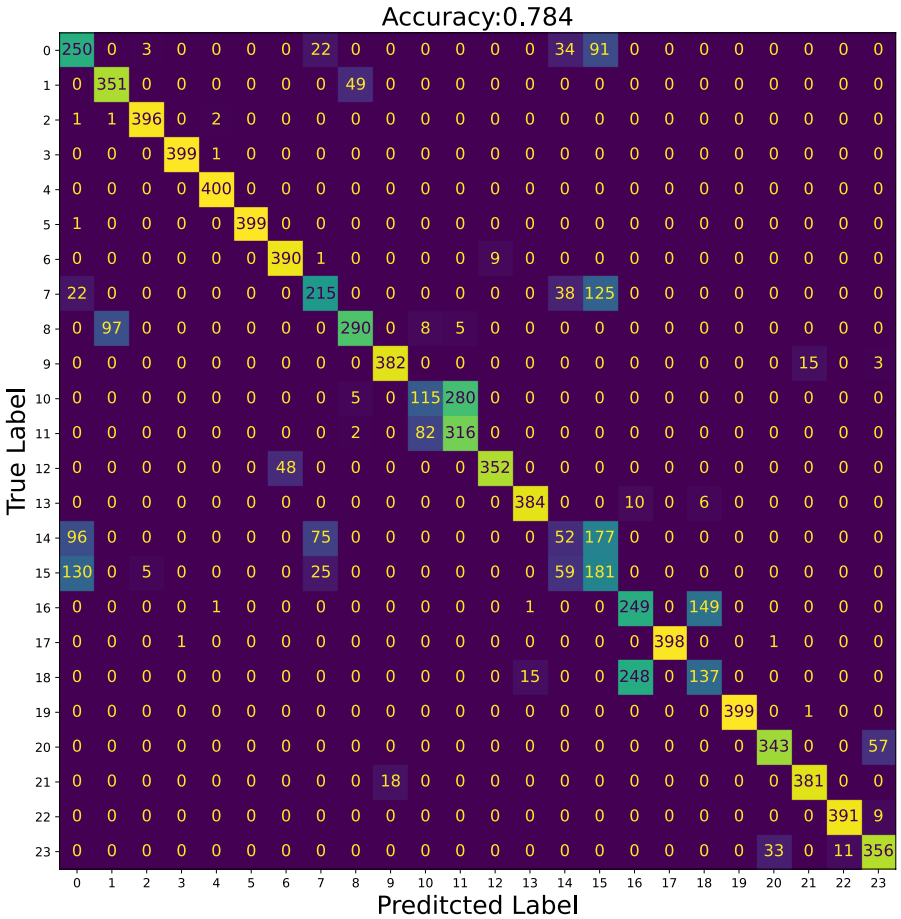


- 24 classes (24 flows)
- 2000 samples per class (power, EM)
- Simulated 2 types of attack
 - Code injection (5-10 NOP instructions)
 - Data exfiltration (through UART pins)
- We trained 2 different classifiers
 - 1 Based on EM signals
 - 1 Based on power signals

Results - classification

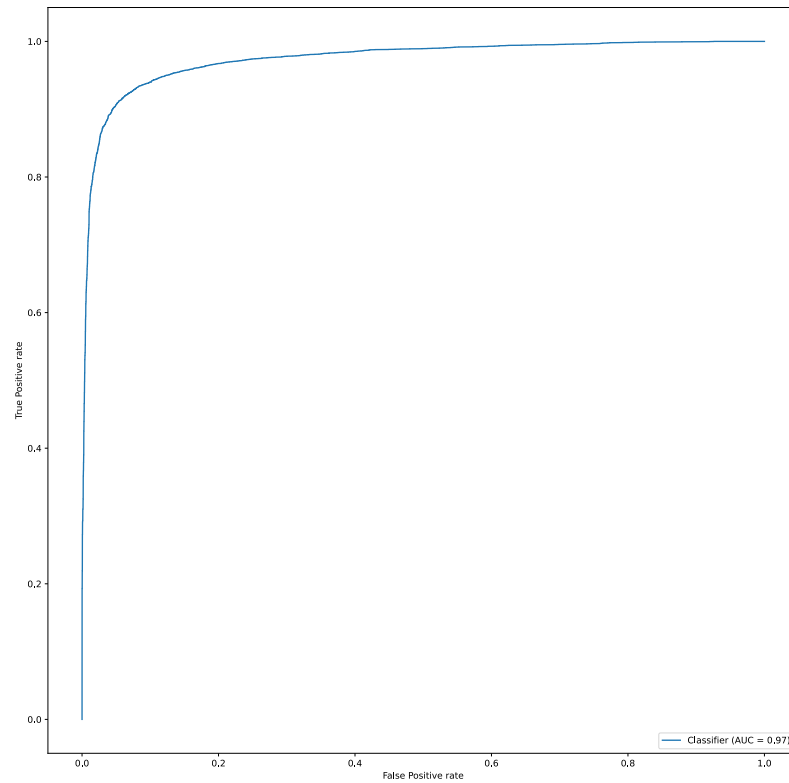


EM Based model

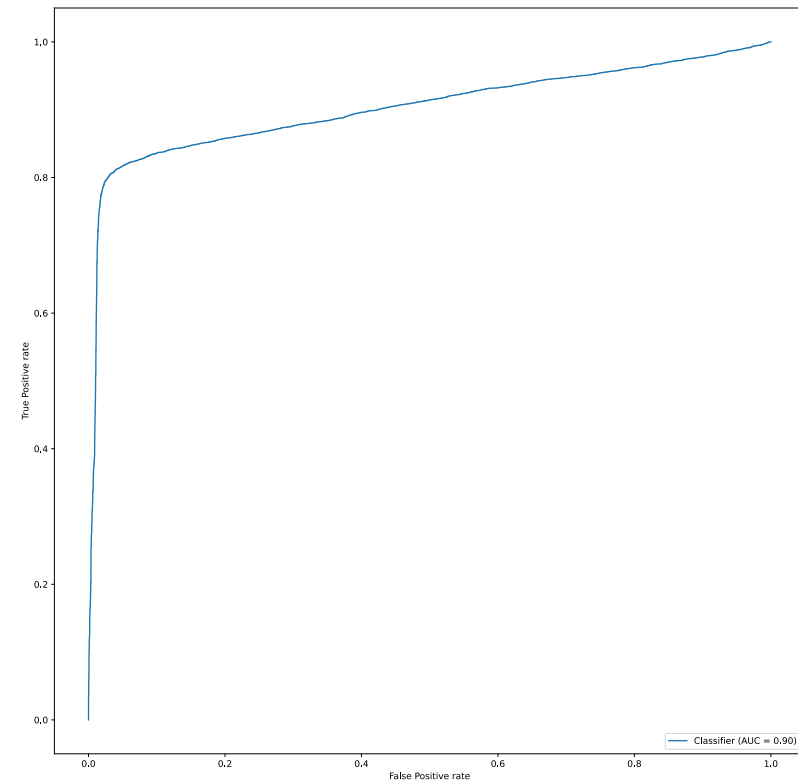


Power Based model

Results – anomaly detection



EM based model

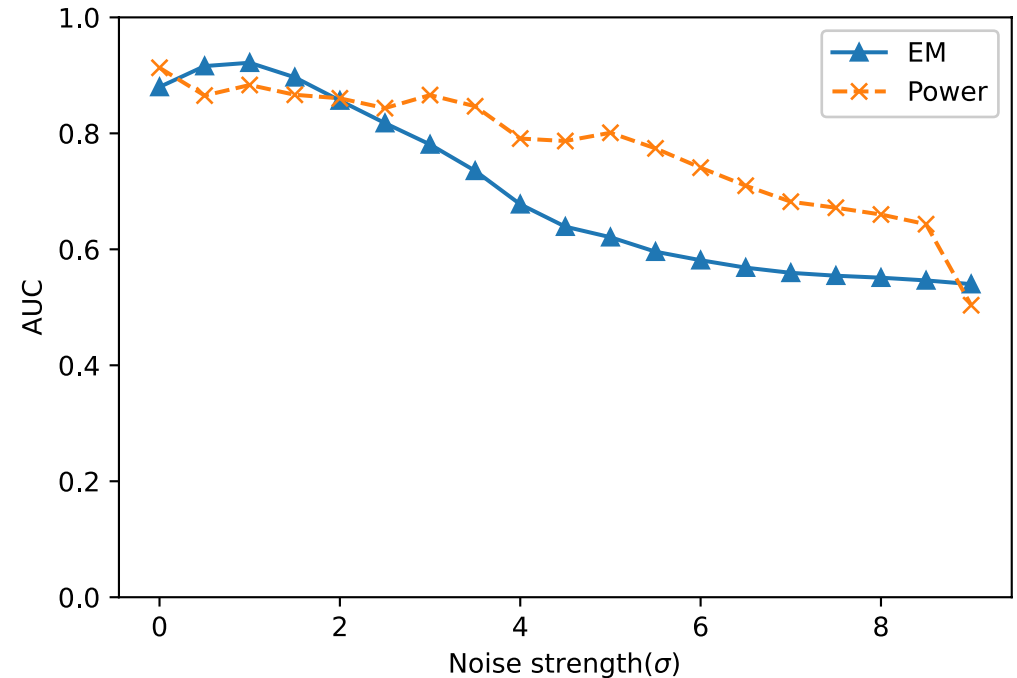
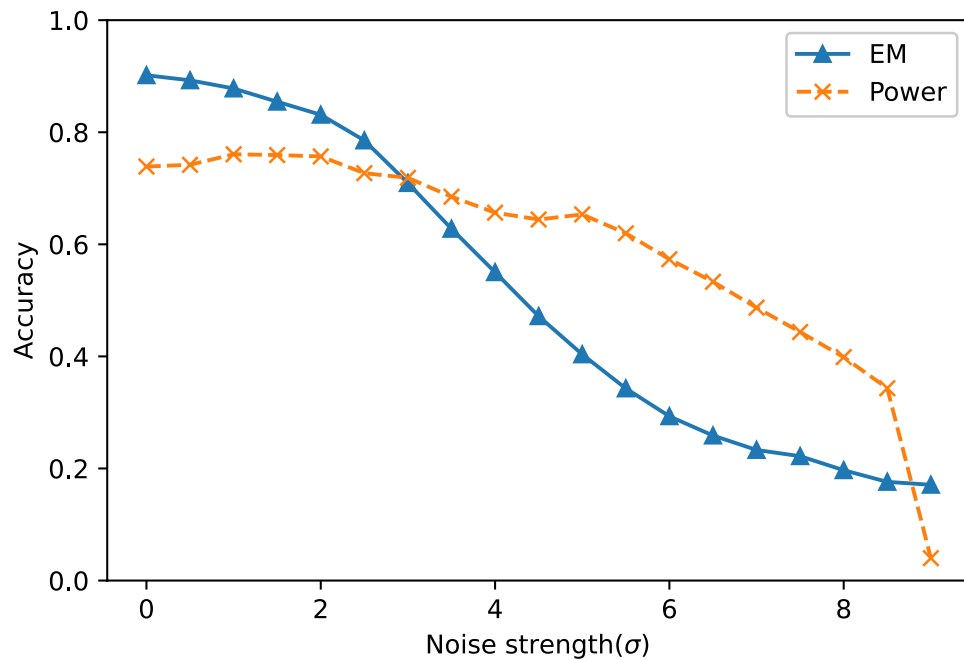


Power based model

Effect of noise



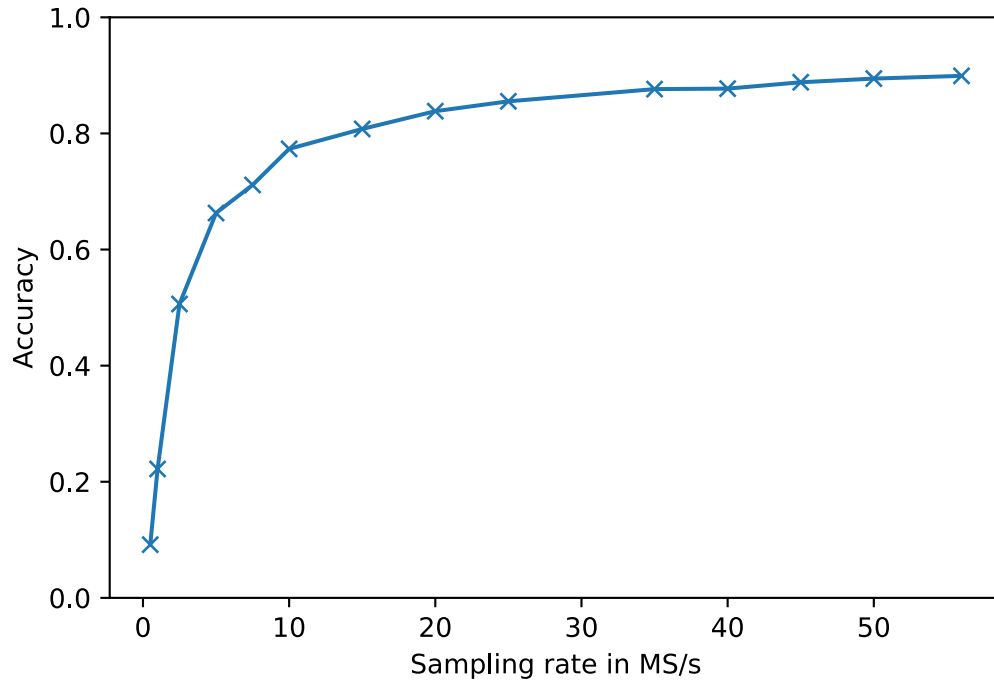
- Samples are collected in nearly optimal environment
- In reality, noise is present



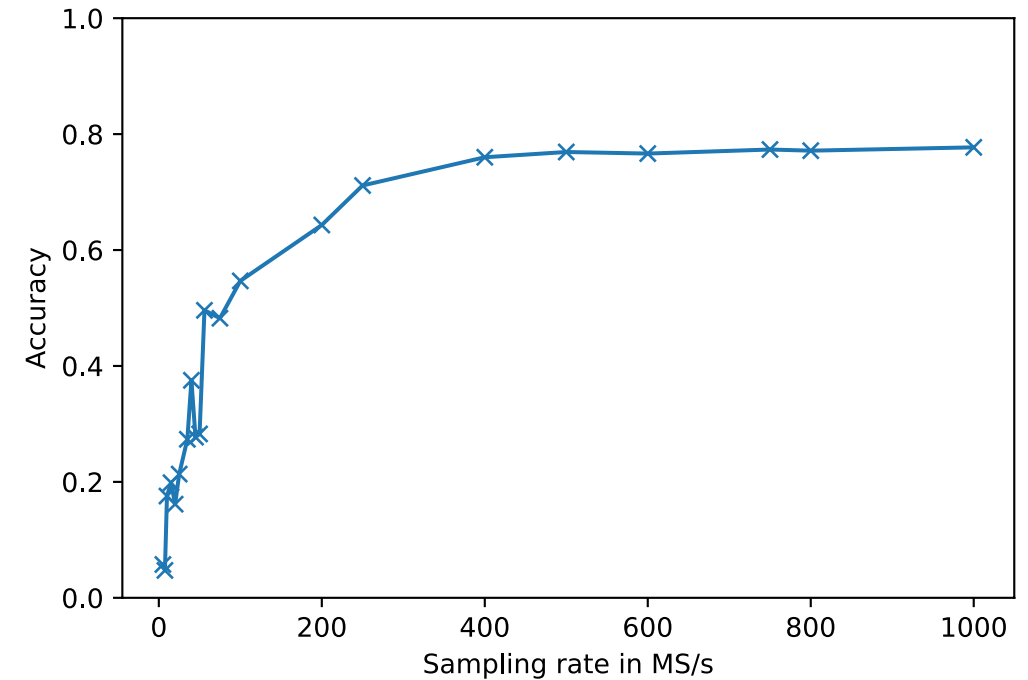
Effect of the sampling rate



- How much can we reduce the sampling rate?



EM based model



Power based model

Choosing the output layer

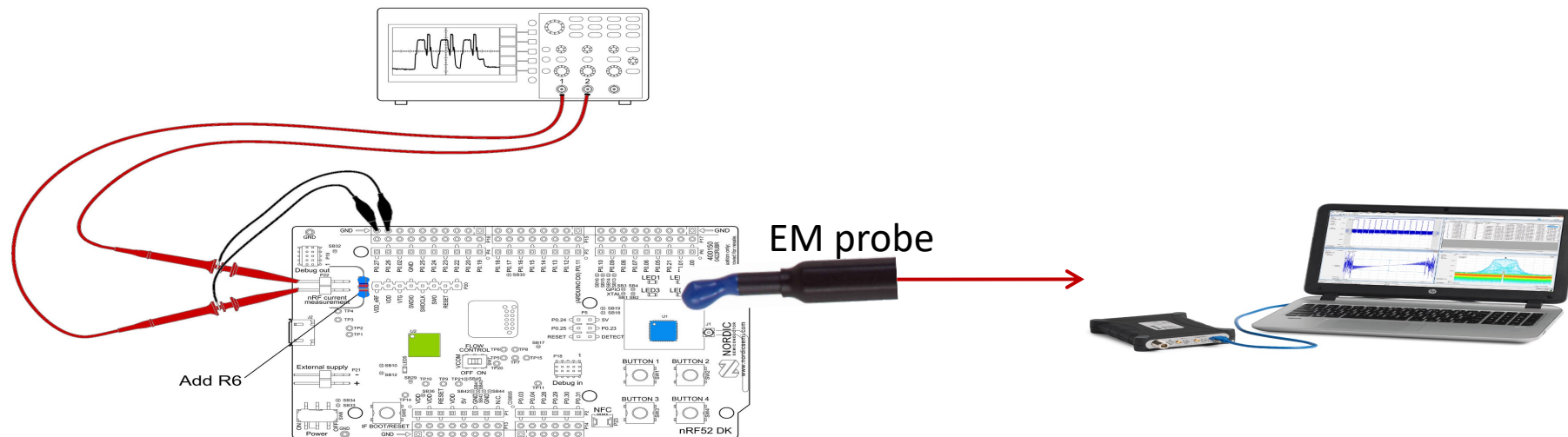


- Usually, SoftMax is used for classification
 - Returns a distribution function:
 - Each output value is $[0,1]$
 - Sum output vector is 1
 - Example: $[0.4, 0.5, 0.1]$
- An increase for 1 class is a decrease for another
- In our use case, low confidence in all classes is desired: $[0.1, 0.1, 0.1]$
- → We use sigmoid as the final output layer
 - Mostly used for multilabel classification
 - Each output value is $[0,1]$
 - Sum output vector is not 1

Cartography



- Quality of the EM signal depends on several variables:
 - Capture frequency
 - Location of the EM probe
- We want to optimize those variables

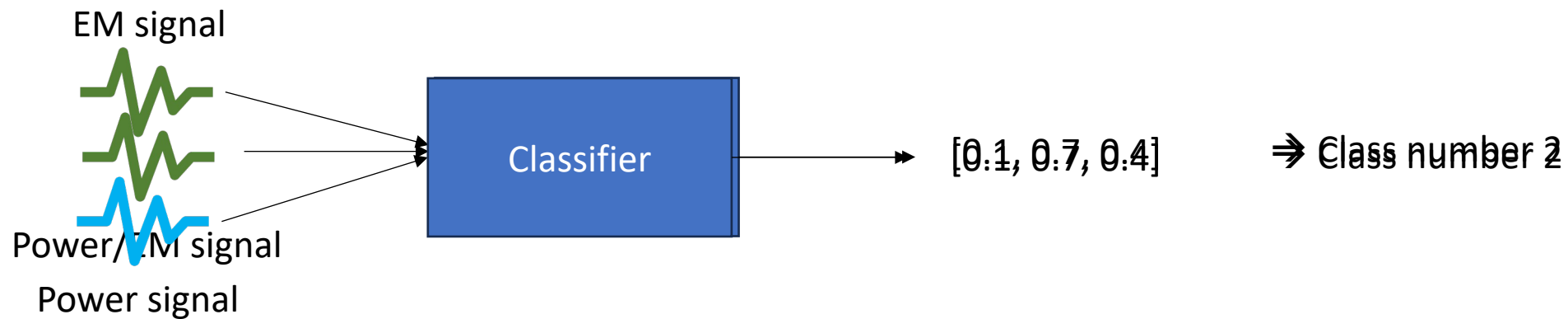


Future work



- We have 2 different models
 - EM based model
 - Power consumption-based model

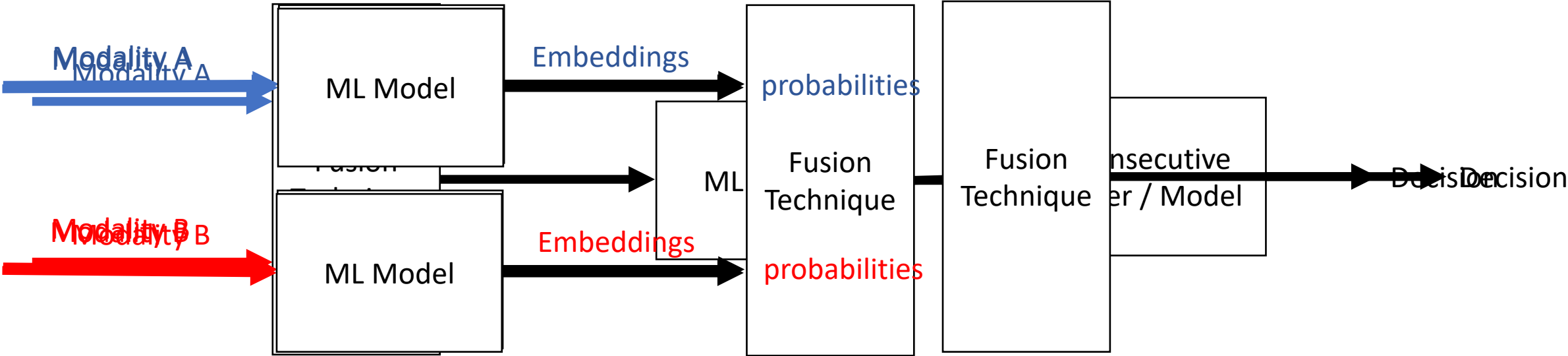
- Why not multimodal?



Multimodal approaches



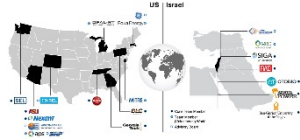
- Early/intermediate/ratio integration



Commercialization



Thank You!



orenlab.sise.bgu.ac.il



twitter.com/amarmic96