ASSURED

Cybersecurity and Insider Threats

GNNs-Based Zero-Assumption Control-Flow Attestation

Marco Chilese. Richard Mitev

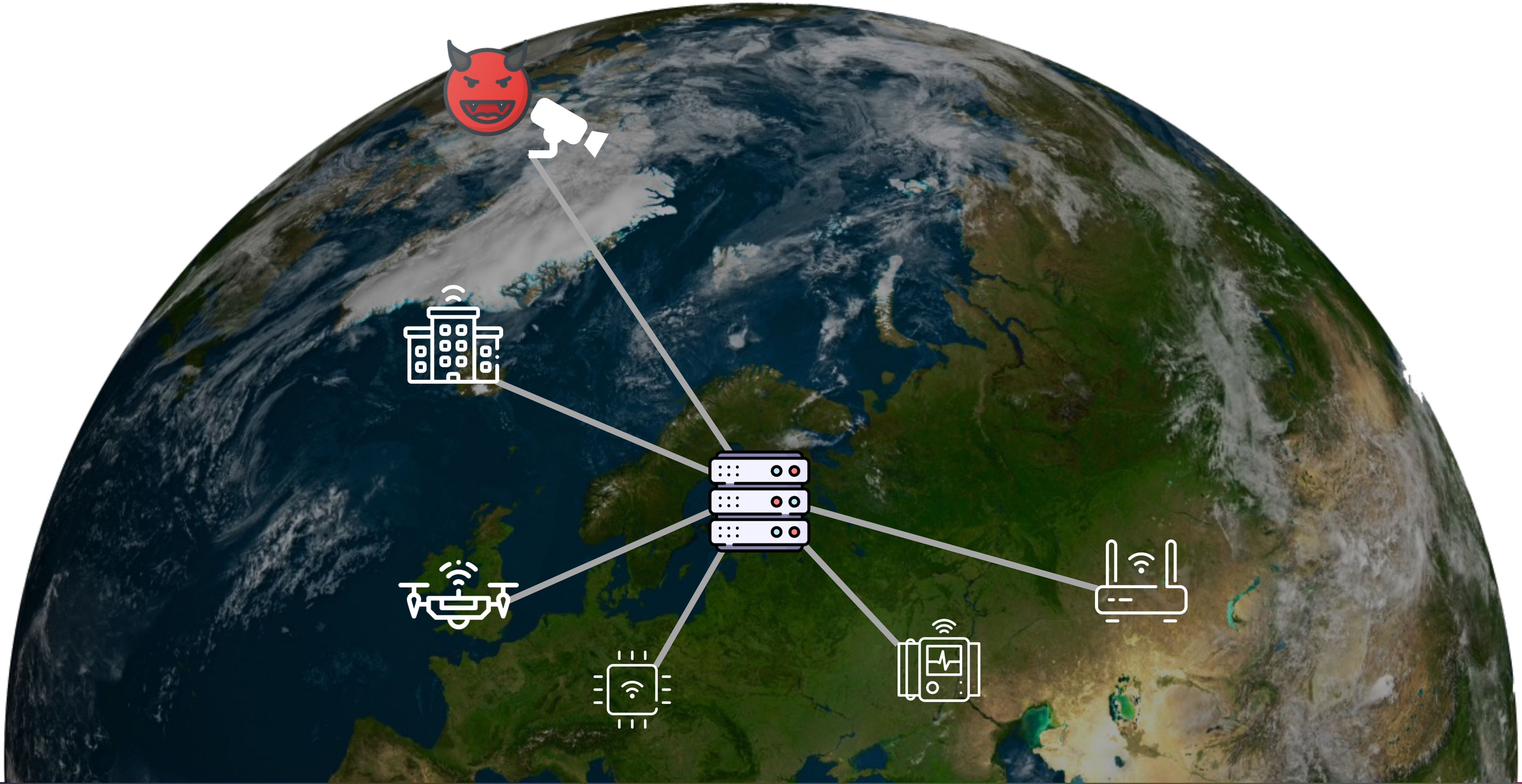
Technical University of Darmstadt, Germany

Webinar

Online, 31.05.2023

Why Attestation?

ASSURED



Motivation

- Current approaches have **unrealistic assumptions**
  - Full Control-Flow Graph
  - Memory access
  - Custom hardware

- In real-world we cannot rely on any of these assumptions
  - **Extracted CFGs are incomplete → Machine Learning?**

Research Question

Is it possible to utilize Machine Learning on incomplete CFG for Control Flow Attestation?

Related Work

ASSURED

Supervised Neural Networks for the Classification of Structures [Sperduti et al., IEEE Transactions on NNs, 1997]

Attest Android software by using API calls and RNN LSTM model [Ma et al., IEEE Access, 2019]

Identifies malicious CF subgraphs of malware executions with GNNs [Herath et al., DSN, 2022]

1995 | 2000 | 2005 | 2010 | 2015 | 2020 | 2025

Variational Graph Auto-Encoders [Kipf et al., ArXiv, 2016]

ML-supported CFA [Hu et al., Trust-Com/BigDataSE, 2019]

Verify state of devices by classifying its memory [Aman et al., IoT Journal, 2022]

State of the Art

ASSURED

Ma et al., IEEE Access, 2019	Limited Information
Hu al., Trust-Com, 2019	Labelled dataset, overhead
Herath et al., DSN, 2022	Labelled dataset, no detection
Aman et al., IoT Journal, 2022	Privacy nightmare

State of the Art – Not only ML

ASSURED

Software Instrumentalization

C-Flat
[Abera *et al., ACM SIGSAC, 2016*]

RECFA
[Zhang *et al.*, ACSAC, 2021]

Extra Hardware

Lo-fat
[Dessouky *et al.*, DAC, 2017]

Atrium
[Zeitouni *et al.*, ICCAD, 2017]

Tiny-CFA
[De Oliveira Nunes *et al.*, DATE, 2021]

- **In general**: large database of full execution path or CF events

State of the Art – Not only ML

Our approach

- NO Full CFG
- NO memory access
- NOT platform specifc
- NO extra HW
- NO data labelling

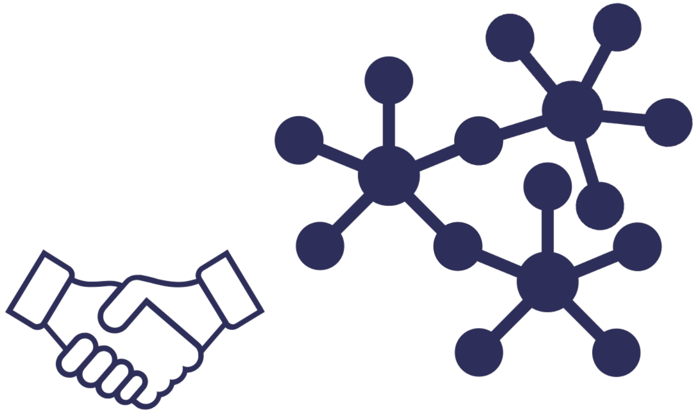
- Lightweight
- 1 Benign CF
- Privacy Enhancing
- ROP Detection
- DOP Detection

State of the Art – Not only ML

Our approach

- NO Full CFG
- NO memory access
- NOT platform specifc
- NO extra HW
- NO data labelling

- Lightweight
- 1 Benign CF
- Privacy Enhancing
- ROP Detection
- DOP Detection

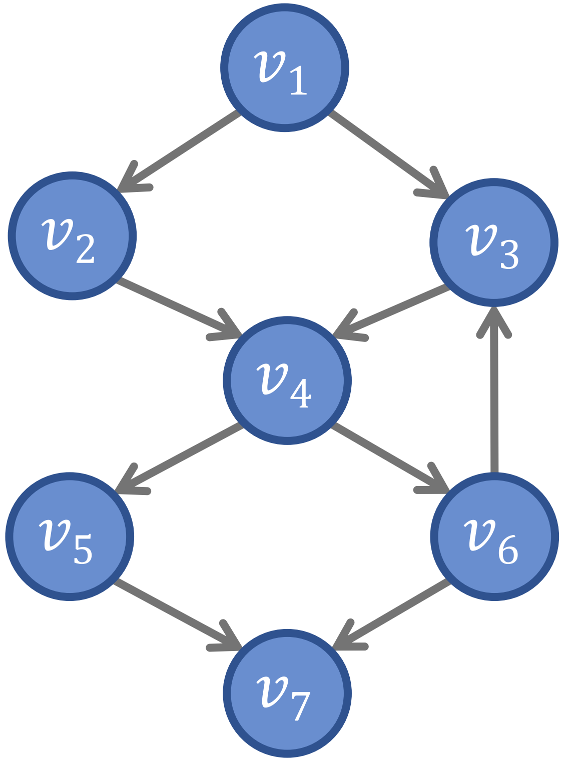


Unsupervised Graph Neural Networks

Background – CFA Attacks

ASSURED

- Two different benign executions in the CFG
- A **ROP attack** adds **extra transitions**
- A **DOP attack reuses benign transitions** but still alters the Control Flow

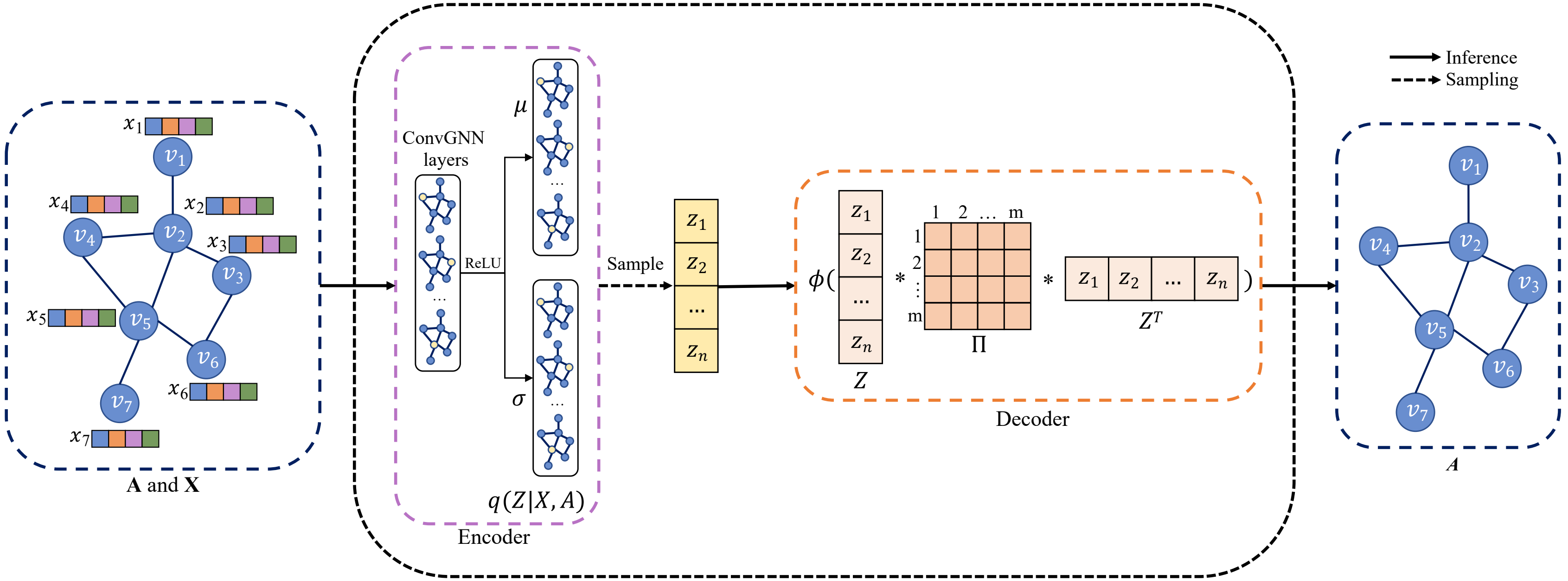


Control Flow Graph
(CFG)

Background – VGAEs

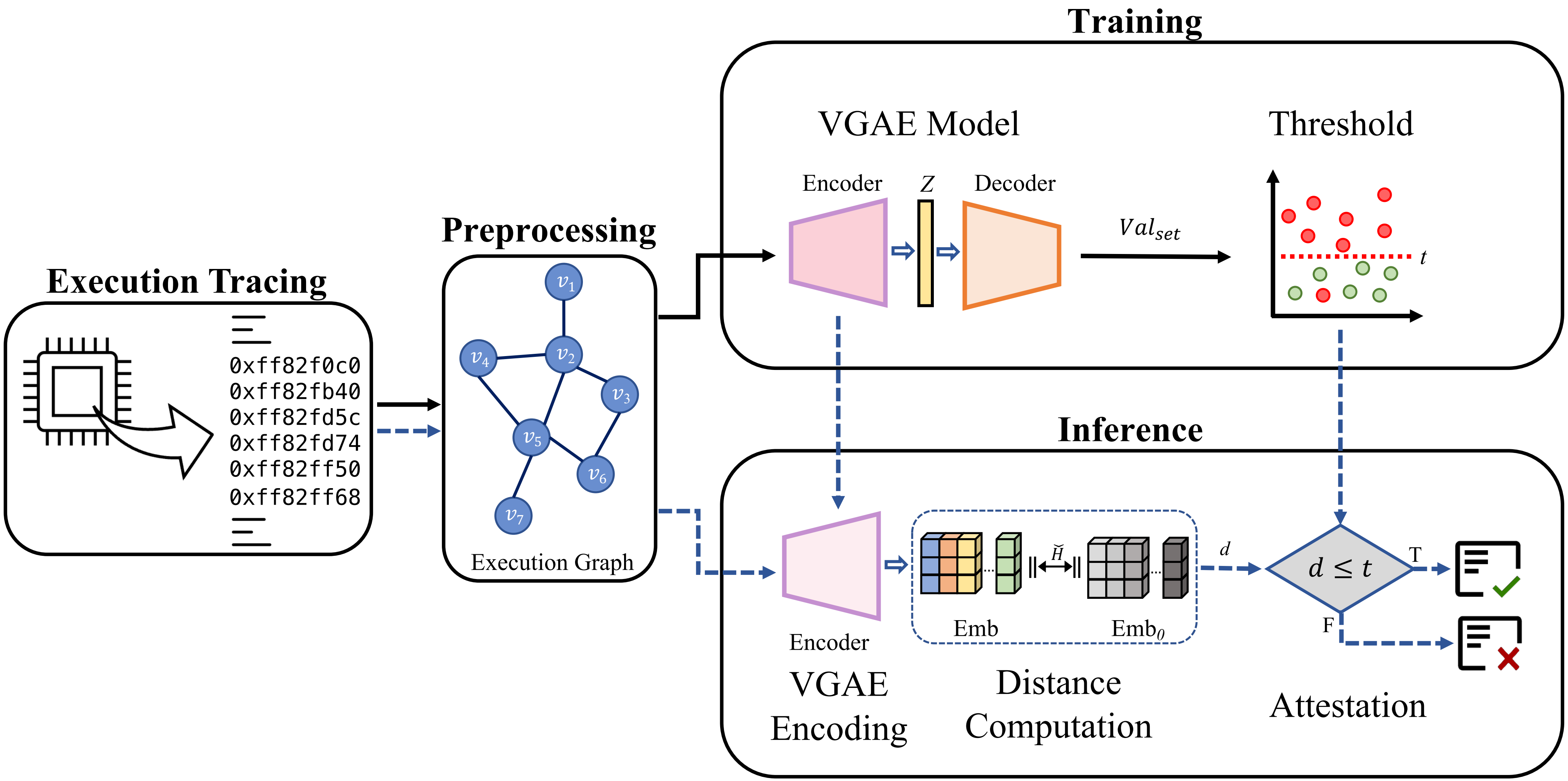
ASSURED

- VGAEs are SotA **generative models** that are trained to learn to reconstruct input graphs
  - Doing so it learns to capture **latent features** of the graph



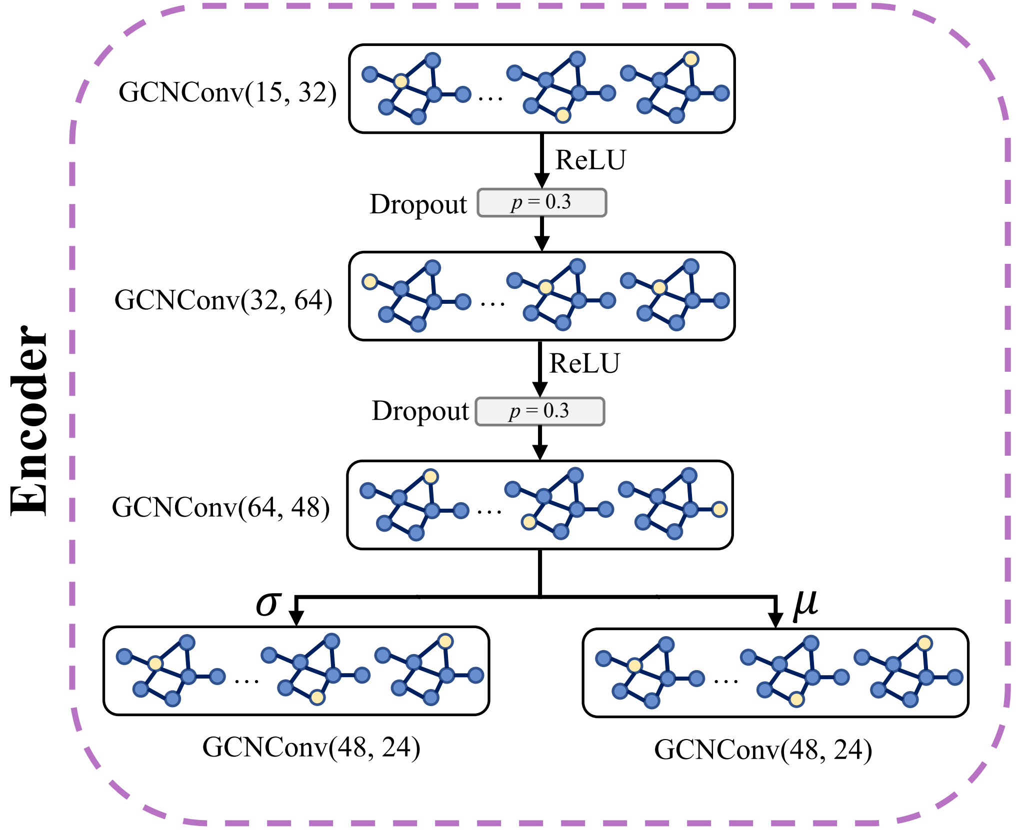
System Overview

ASSURED



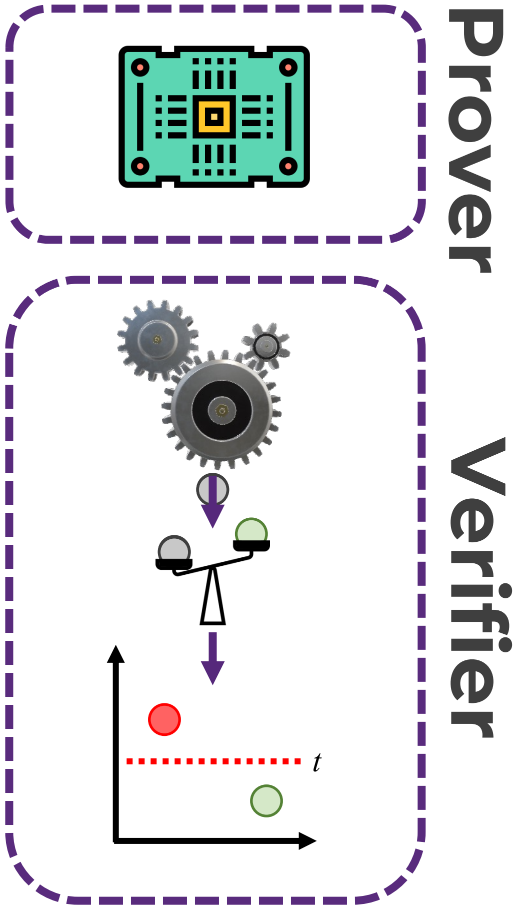
Implementation – VGAE Model

- From the encoder we obtain the **nodes' embeddings**
  - We can imagine them as a "fingerprint" of each node that is capturing its characteristics
- We designed our **deep Encoder** so to capture graphs' characteristics (e.g., connectivity, neighbours)
- **Regularization** through dropout layers and custom **decaying learning rate** schema
- The model counts **only 8,128 parameters**



Implementation - Attestation

- **Directed Hausdorff distance** between executions
- Attestation through **threshold test**
  - Calibrated on small benign validation set
- **Attack independent**



Evaluation

ASSURED

- Average F1-Score
  - ROP Attacks: 97.71%
  - DOP Attacks: 86.52%
- Average False-Positive-Rate of 3.63%

ROP Dataset	FPR	Pr.	Re.	F1
Diffie-Hellman	3.30	94.5	89.17	91.76
DES	7.72	96.93	99.37	98.14
DESX	7.34	97.08	99.22	98.14
GOST	1.16	99.53	99.53	99.53
AES	0.39	99.84	98.74	99.29
EmbenchIoT*	1.85	99.72	99.16	99.42
⌀	3.63	97.93	97.53	97.71

DOP Dataset	FPR	Pr.	Re.	F1
Diffie-Hellman	3.30	70.27	78.00	73.93
DES	7.72	83.33	100.00	90.90
DESX	7.34	83.90	99.00	90.83
GOST	1.16	94.00	70.15	80.34
AES	0.39	97.61	77.36	86.31
EmbenchIoT*	1.85	95.5	98.6	96.8
⌀	3.63	87.44	87.19	86.52

*EmbenchIoT includes 18 software

Evaluation: Performance

Prover

- **Average Trace Steps:** 6.5×10^6
- **Preprocessing Time:** ~1-3s*
- **Processed Trace Size:** 0.94KB

Verifier

- **Model Training Time:** 5min*
- **Attestation Runtime:** 0.005s*

*Timings observed on Raspberry Pi 4B 2GB

THANKS

PROJECT-ASSURED.EU

@Project_Assured

ASSURED project is funded by the EU's Horizon2020
programme under Grant Agreement number 952697