Lab-STICC
UMR 6285

Hardware/Software Security

Yehya NASSER
Associate Professor

Lab-STICC
IMT ATLANTIQUE
Brest France

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Hardware Trojan ML Based Security Model for DNN Accelerators (FALCON)

2026-01-16

PhD Student: Abdel Salam BALTAGI
Supervisors: Yehya NASSER, Associate professor, Principle Investigator (PI)
Amer BAGHDADI, Professor, Co-PI, Thesis Director

Outline

ML-Based Hardware Trojan Detection in AI Accelerators via Power Side-Channel Analysis

AbdelSalam Baltagi, Yehya Nasser, Amer Baghdadi
IMT Atlantique, Lab-STICC, UMR CNRS 6285, F-29238 Brest, France

***Abstract*—To accelerate development and system integration, many companies opt to outsource the design of complex AI accelerators to third-party IP vendors rather than developing them in-house. This practice raises security concerns, particularly the risk of hardware Trojan (HT) attacks [1]. Traditional testing methods are impractical for detecting HTs in modern AI/ML accelerators due to their hardware complexity and inability to provide insights into the inserted Trojans. In this work, we propose a methodology to detect the presence of HTs in different AI/ML accelerators and identify key Trojan characteristics using power side-channel analysis (PSCA). We present a testbed for accurate power consumption measurement, enabling the collection of real ML inference power traces. We inserted multiple HTs in several AI/ML accelerators and prepared various dataset configurations to analyze multiple HT scenarios. Then, we proposed a novel method for preprocessing PSCA data that consists of segmenting the power traces and extracting statistical features from time and frequency domains. Our proposed technique, equipped with an ML-based HT detection and identification method, achieves up to 99% accuracy.**

***Index Terms*—Hardware Trojan, AI Accelerator, Machine learning, HT detection, Power side-channel analysis**

I. Introduction

As AI-specialized hardware (HW), particularly neural network accelerators, moves from academic research to industrial deployment, concerns about its security are receiving increasing attention [1]. In modern applications, convolution neural networks (CNNs) are widely used in tasks such as image classification, face recognition, object detection, and medical diagnosis [2]. Due to their substantial parameter sizes, often exceeding hundreds of megabytes, each inference involves billions of operations, leading to long inference times. This challenge is especially critical in applications requiring fast, accurate responses, such as face recognition at airport security gates and speed sign detection in self-driving vehicles. To

Threats to AI accelerators, presented in Fig. 1, can be categorized along two dimensions: attack category (unintended vs. intended) and attack source (hardware vs. software) [6]. In an unintended attack, the neural network system remains benign, with no modifications or malicious functions, and attackers lack privileges to alter the system's internal state. In contrast, intended attacks compromise the robustness of embedded neural networks, leading to functionality changes, data leakage, or denial-of-service (DoS) attacks. These attacks often involve the insertion of malicious hardware or software (SW) backdoors into CNN accelerators and models, a scenario frequently encountered when outsourcing design and manufacturing to third-party vendors.

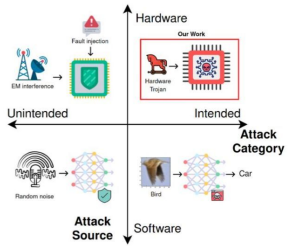


Fig. 1. AI/ML accelerator threats categorization along two dimensions: attack category and attack source

Abstract
Authors
Keywords
Metrics

Abstract:
To accelerate development and system integration, many companies opt to outsource the design of complex AI accelerators to third-party IP vendors rather than developing them inhouse. This practice raises security concerns, particularly the risk of hardware Trojan (HT) attacks [1]. Traditional testing methods are impractical for detecting HTs in modern AI/ML accelerators due to their hardware complexity and inability to provide insights into the inserted Trojans. In this work, we propose a methodology to detect the presence of HTs in different AI/ML accelerators and identify key Trojan characteristics using power side-channel analysis (PSCA). We present a testbed for accurate power consumption measurement, enabling the collection of real ML inference power traces. We inserted multiple HTs in several AI/ML accelerators and prepared various dataset configurations to analyze multiple HT scenarios. Then, we proposed a novel method for preprocessing PSCA data that consists of segmenting the power traces and extracting statistical features from time and frequency domains. Our proposed technique, equipped with an ML-based HT detection and identification method, achieves up to 99% accuracy.

Published in: IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (Early Access)

Page(s): 1 - 1

DOI: 10.1109/TCAD.2025.3625109

Date of Publication: 23 October 2025

Publisher: IEEE

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

LPSS
In Silicon, We Innovate

Outline

1. Context and Background
2. Literature Review
3. Proposed Approaches and Results
4. Conclusion

Outline

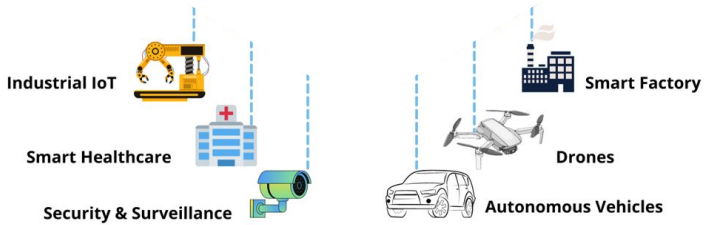
1. **Context and Background**
2. Literature Review
3. Proposed Approaches and Results
4. Conclusion

AI’s Popularity In Embedded Systems

AI applications are getting more popular

Implemented on embedded systems (SoC-FPGAs)

Autonomous vehicles, military, IoT, and control systems



[1] N.Bhatta, "**Machine Learning-Based Classification of Hardware Trojans Using Power Side-Channel Signals**," IEEE 67th International Midwest Symposium on Circuits and Systems (MWSCAS) - **2024**

AI's Popularity In Embedded Systems

AI applications are getting more popular

Implemented on embedded systems (SoC-FPGAs)

Autonomous vehicles, military, IoT, and control systems

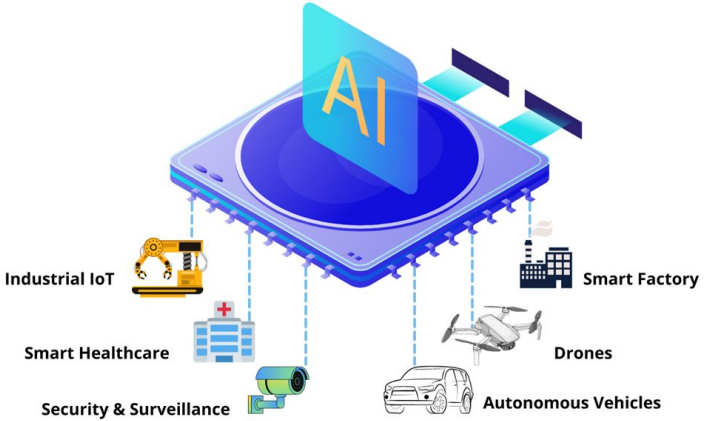
Implemented on Complex HW

Building the whole system in-house is challenging

Difficulty noticing malicious circuits added by a third party

Security of the HW Platforms is Crucial

Especially for critical applications [1]



[1] N.Bhatta, "**Machine Learning-Based Classification of Hardware Trojans Using Power Side-Channel Signals**," IEEE 67th International Midwest Symposium on Circuits and Systems (MWSCAS) - **2024**

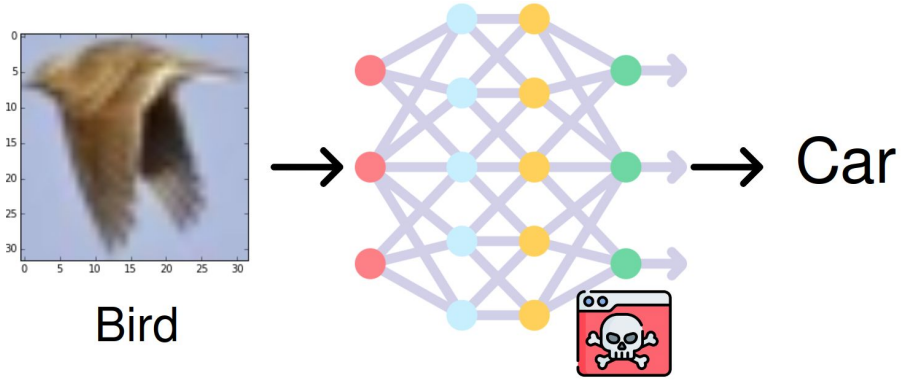
Vulnerability of Hardware Attacks

- **Hardware Attacks [1]**
  - Fault injection
  - Side-channel attack
  - Hardware Trojan (HT)

[1] Q. Xu, M. Tanvir Arafin and G. Qu, "**Security of Neural Networks from Hardware Perspective: A Survey and Beyond**," **2021** 26th Asia and South Pacific Design Automation Conference (ASP-DAC), 2021, pp. 449-454
[2] K. I. Gubbi, "**Hardware Trojan Detection Using Machine Learning: A Tutorial**," ACM Transactions on Embedded Computing Systems, **2023**

Vulnerability of Hardware Attacks

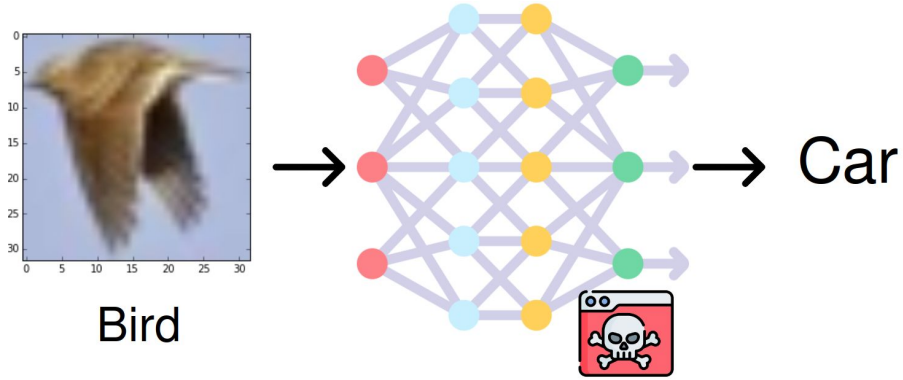
- **Hardware Attacks [1]**
  - Fault injection
  - Side-channel attack
  - **Hardware Trojan (HT)**



[1] Q. Xu, M. Tanvir Arafin and G. Qu, "**Security of Neural Networks from Hardware Perspective: A Survey and Beyond**," **2021** 26th Asia and South Pacific Design Automation Conference (ASP-DAC), 2021, pp. 449-454
[2] K. I. Gubbi, "**Hardware Trojan Detection Using Machine Learning: A Tutorial**," ACM Transactions on Embedded Computing Systems, **2023**

Vulnerability of Hardware Attacks

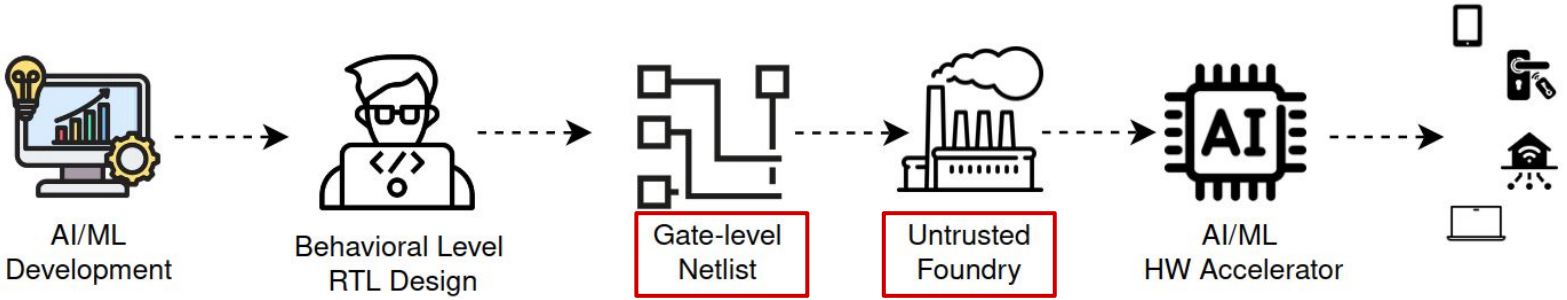
- **Hardware Attacks [1]**
  - Fault injection
  - Side-channel attack
  - **Hardware Trojan (HT)**



- **Why ML for HT Detection? [2]**
  - Today's ICs are huge and complex
  - HTs are minute and difficult to detect
  - Finding the HT early in the design stage is important
  - Reducing HT detection cost

[1] Q. Xu, M. Tanvir Arafin and G. Qu, "**Security of Neural Networks from Hardware Perspective: A Survey and Beyond**," **2021** 26th Asia and South Pacific Design Automation Conference (ASP-DAC), 2021, pp. 449-454
[2] K. I. Gubbi, "**Hardware Trojan Detection Using Machine Learning: A Tutorial**," ACM Transactions on Embedded Computing Systems, **2023**

Hardware Trojan Threats and Potential Mitigations



Outline

1. Context and Background
2. **Literature Review**
3. Proposed Approaches and Results
4. Conclusion

HT Detection Recent Work

HW Monitor

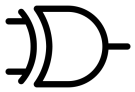
Hardware overhead
Limited observability

[1] J.Hou, "**Hardware Trojan Attacks on the Reconfigurable Interconnections of Field-Programmable Gate Array-Based Convolutional Neural Network Accelerators and a Physically Unclonable Function-Based Countermeasure Detection Technique**," - **2024**
[2] Z.Liu, "**Novel Two-Level Protection Scheme against Hardware Trojans on a Reconfigurable CNN Accelerator**," - **2024**
[3] H.Esmaeili, "**A Pre-Activation, Golden IC Free, Hardware Trojan Detection Approach**," - **2022**

HT Detection Recent Work

HW Monitor

Hardware overhead
Limited observability



Netlist Analysis

Not always accessible
Can be inaccurate or falsified

[4] S. Takemoto, "**AI Hardware Oriented Trojan Detection Architecture**," - **2022**
[5] D.C, "**A machine-learning-based hardware-Trojan detection approach for chips in the Internet of Things**," - **2019**

HT Detection Recent Work

HW Monitor

Hardware overhead
Limited observability

Netlist Analysis

Not always accessible
Can be inaccurate or falsified

Side-Channel Analysis

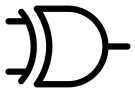
Quick and easy to acquire
Reflects real hardware behavior

[6] N.Bhatta, "**Machine Learning-Based Classification of Hardware Trojans Using Power Side-Channel Signals**," - **2024**
[7] A.Nasr, "**A Siamese deep learning framework for efficient hardware Trojan detection using power side-channel data**," - **2024**
[8] L.Liao, "**Hardware Trojan Detection and Identification Using Electromagnetic Side-Channel Leakage**," - **2024**
[9] K. I. Gubbi, "**Hardware Trojan Detection Using Machine Learning: A Tutorial**," - **2023**
[10] A. Roy, "**Detection of Hardware Trojan using CNN with Residual Network**," - **2022**.
[11] S. Yang, "**Side-channel Analysis for Hardware Trojan Detection using Machine Learning**," - **2021**

HT Detection Recent Work

HW Monitor

Hardware overhead
Limited observability



Netlist Analysis

Not always accessible
Can be inaccurate or falsified

Side-Channel Analysis

Quick and easy to acquire
Reflects real hardware behavior

Targeted cryptographic hardware

FALCON Project Objective

Hardware Trojan **ML Based Security Model for DNN Accelerators**

Hardware Attack

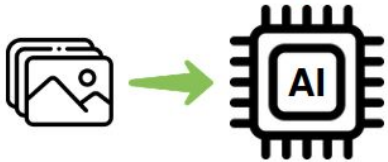
Detection based on ML Techniques

Targeted for AI/CNN Accelerators

Outline

1. Context and Background
2. Literature Review
3. **Proposed Approaches and Results**
   a. **HT Presence Detection**
   b. HT Impact Identification
4. Conclusion

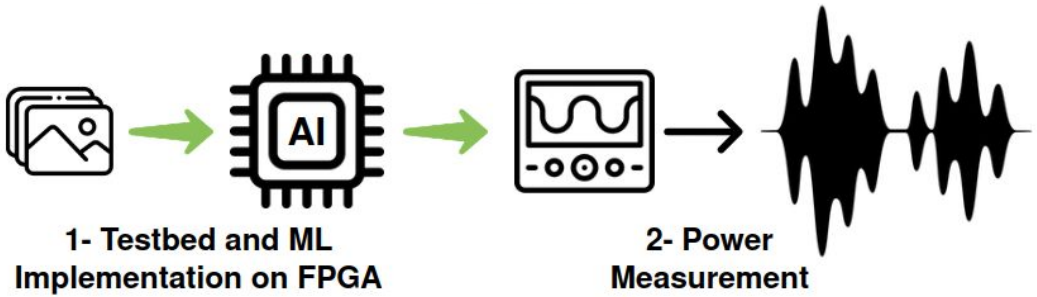
Proposed Approach*



1- Testbed and ML Implementation on FPGA

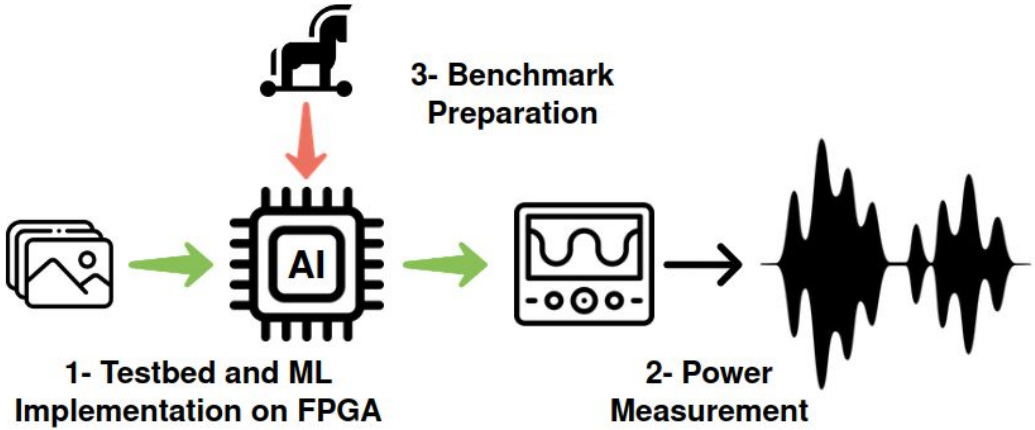
* This section presents our work published in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD): A. Baltagi, Y. Nasser, A. Baghdadi, "**ML-Based Hardware Trojan Detection in AI Accelerators via Power Side-Channel Analysis**, " 2025

Proposed Approach*



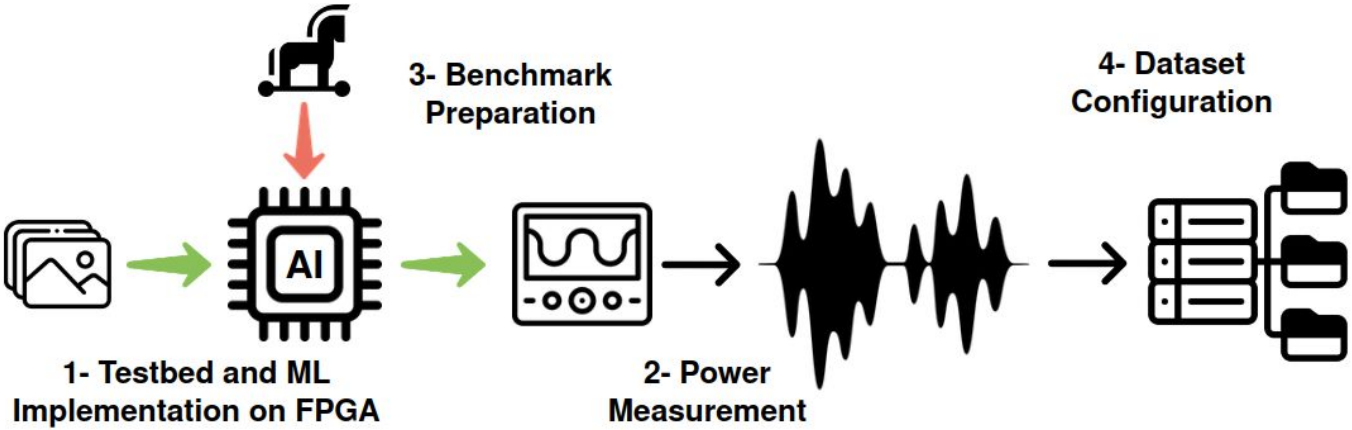
* This section presents our work published in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD): A. Baltagi, Y. Nasser, A. Baghdadi, "**ML-Based Hardware Trojan Detection in AI Accelerators via Power Side-Channel Analysis**, " 2025

Proposed Approach*



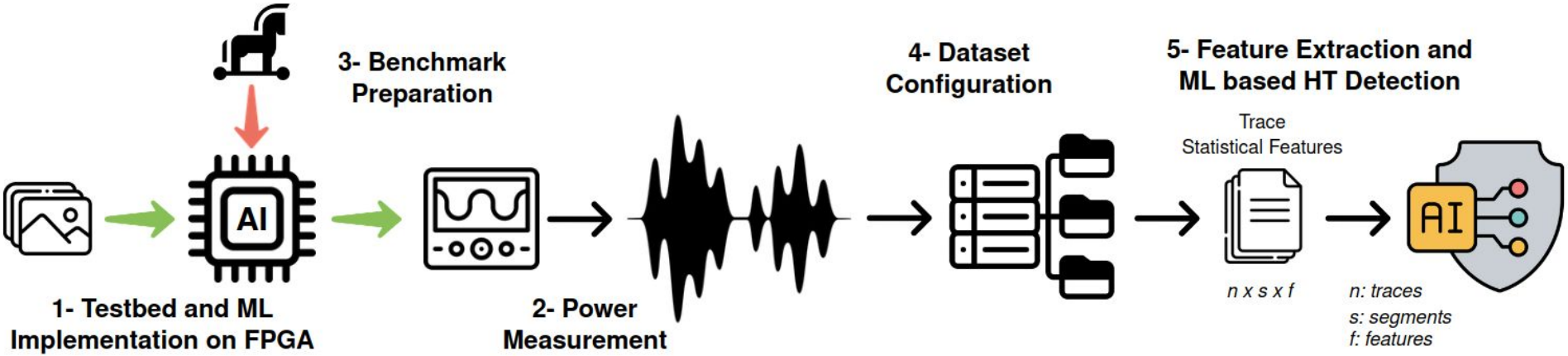
* This section presents our work published in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD): A. Baltagi, Y. Nasser, A. Baghdadi, "**ML-Based Hardware Trojan Detection in AI Accelerators via Power Side-Channel Analysis**, " 2025

Proposed Approach*



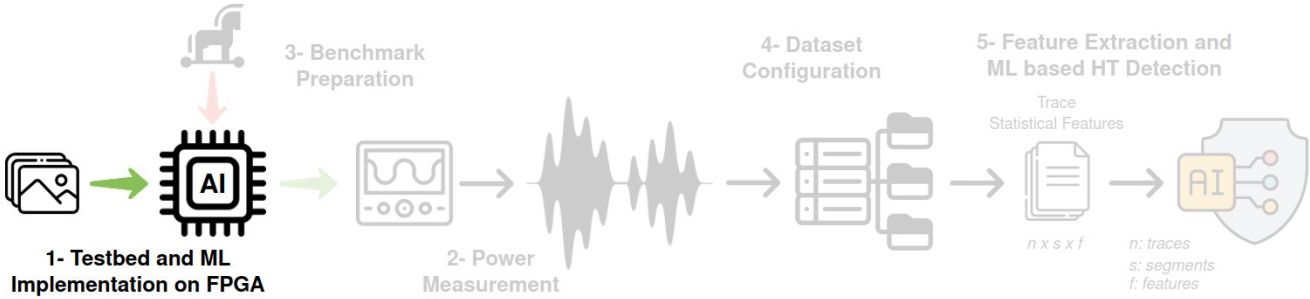
* This section presents our work published in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD): A. Baltagi, Y. Nasser, A. Baghdadi, "**ML-Based Hardware Trojan Detection in AI Accelerators via Power Side-Channel Analysis**, " 2025

Proposed Approach*

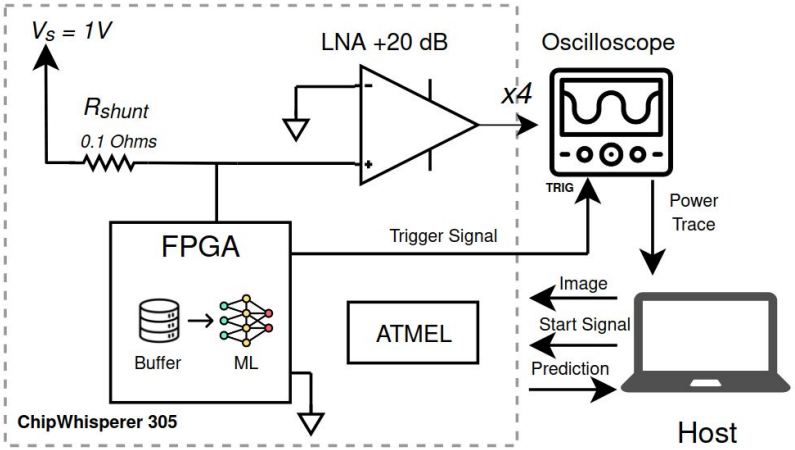


* This section presents our work published in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD): A. Baltagi, Y. Nasser, A. Baghdadi, "**ML-Based Hardware Trojan Detection in AI Accelerators via Power Side-Channel Analysis**, " 2025

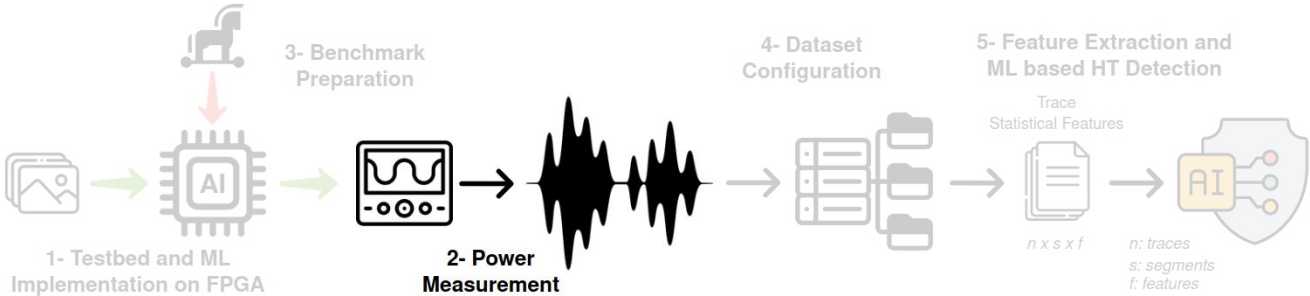
Main Workflow - Step 1: Testbed and ML Implementation on FPGA



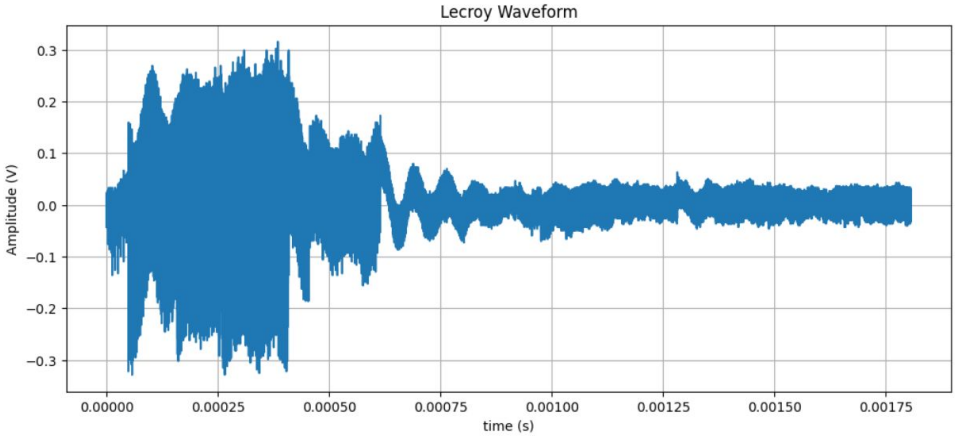
- ChipWhisperer 305 Target Board - Artix 7 FPGA
- Implemented 4 different CNN accelerators
- Teledyne LeCroy Oscilloscope
- Testbed automation using Python script



Main Workflow - Step 2: Power Measurement



- High fluctuation due to parallelization
- Long power trace
  - Cryptographic HW ⟶ 11 cycles
  - AI HW ⟶ 450,000 cycles
- Oversampling rate [1]: $f_{sampling} = 5 \times f_{FPGA}$

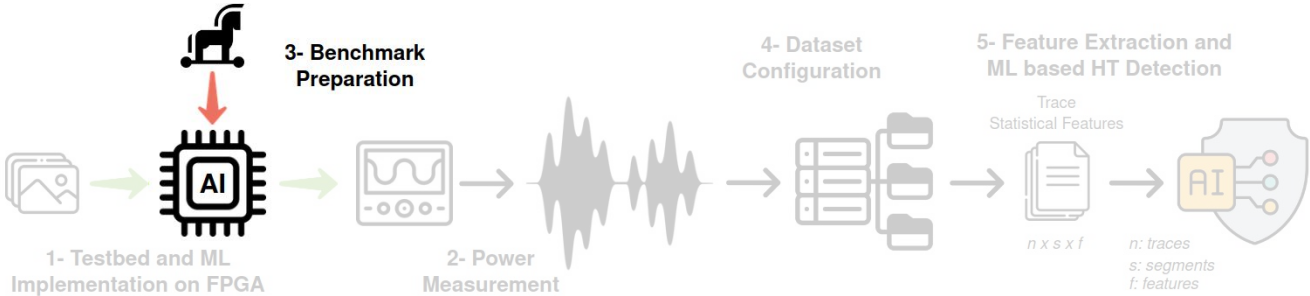


IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

LPSS
In Silicon, We Innovate

[1] Getting The Most Out Of Your Oscilloscope: Setup, "https://blog.teledynelecroy.com/2018/01/getting-most-out-of-your-oscilloscope.html" - 2018

Main Workflow - Step 3: Benchmark Preparation



- Implemented 15 HTs inspired from literature
- Attributes:
  - Trigger type
  - Payload type
  - Always-on
  - Location (Model's Layer)

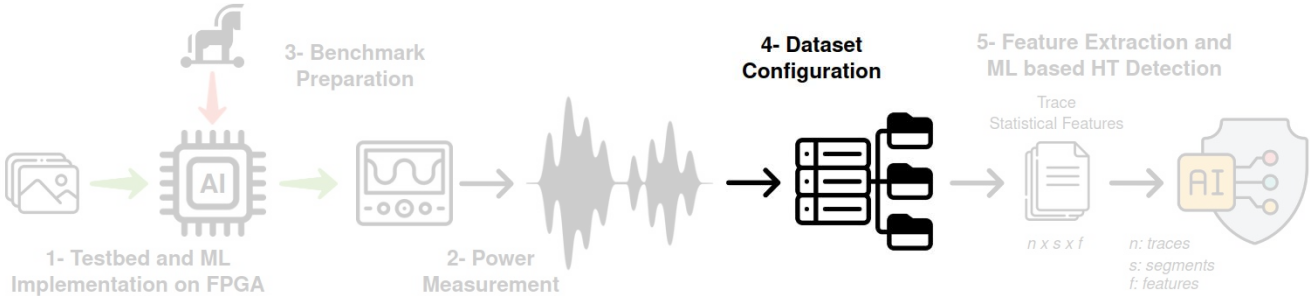
Name	Trigger	Payload	Always-on	Location
ext_comb_2	input pixels	bypass	false	external
ext_seq_4	input pixels	bypass	false	external
relu_fc7	internal sig.	block mod.	false	MVAC_7
ht_shft_10_MsCls	internal sig.	softmax	false	Label Select
ht_shft_5_MsCls	internal sig.	softmax	false	Label Select
ht_shft_1_MsCls	internal sig.	softmax	false	Label Select
ht_MxTgMs	internal sig.	softmax	false	Label Select
ht_SwSTMs	internal sig.	softmax	false	Label Select
maxpool_0_always-on	-	maxpool	true	Maxpool
maxpool_0_pattern-comparator	internal sig.	maxpool	false	Maxpool
maxpool_0_pattern-counter	input pixels	maxpool	false	Maxpool
reconfigurable_mvac_0	internal sig.	block mod.	false	MVAC_0
geometric_trojan	input pixels	bypass	false	external
clock_glitch_always-on	-	clock-glitch	true	Label Select
clock_glitch_pattern-recognition	input pixels	clock-glitch	false	Label Select

IMT Atlantique
Bretagne-Pays de la Loire
Ecole Mines-Télécom

LPSS
In Silicon, We Innovate

* Benchmark references are found in the supporting slides

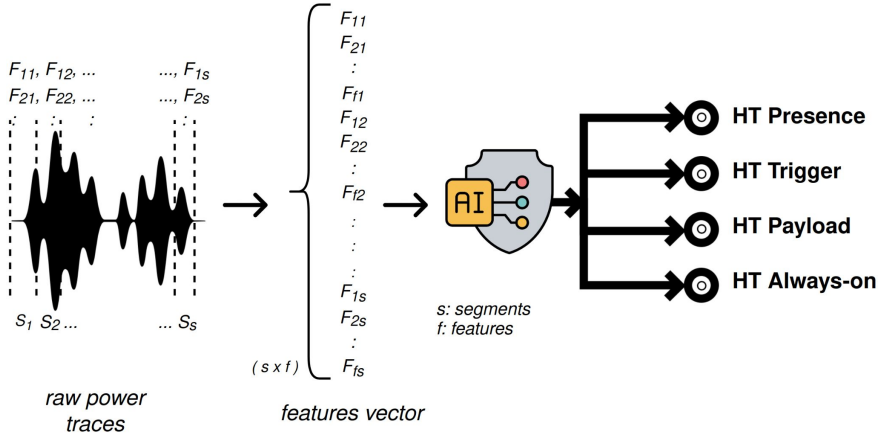
Main Workflow - Step 4: Dataset Configuration



	Configuration	Description	Nb of Classes	Dataset Size (traces)
Binary-class Configurations	Binary-Diff	Clean vs. HTs from different groups	2	50K
	Binary-Bypass	Clean vs. bypass HTs group		30K
	Binary-Block-mod	Clean vs. block modification HTs group		20K
	Binary-Softmax	Clean vs. softmax HTs group		50K
	Binary-Maxpool	Clean vs. maxpool HTs group		30K
	Binary-Clock-Glitch	Clean vs. clock-glitch HTs group		20K
Multi-class Configurations	Multi-All	Class per HW design	16	80K
	Multi-Trigger	Clean vs. trigger groups	3	70K
	Multi-Payload	Clean vs. payload groups	6	90K
	Multi-Always-on	Mixed vs. always-on groups	2	25K

Main Workflow - Step 5: Feature Extraction and ML Based HT Detection

- Slice the power trace
- Extract statistical features per segment
- Goal: Detect and Identify
- Train classifiers using different configurations
- XGBoost, Random Forest, SVM



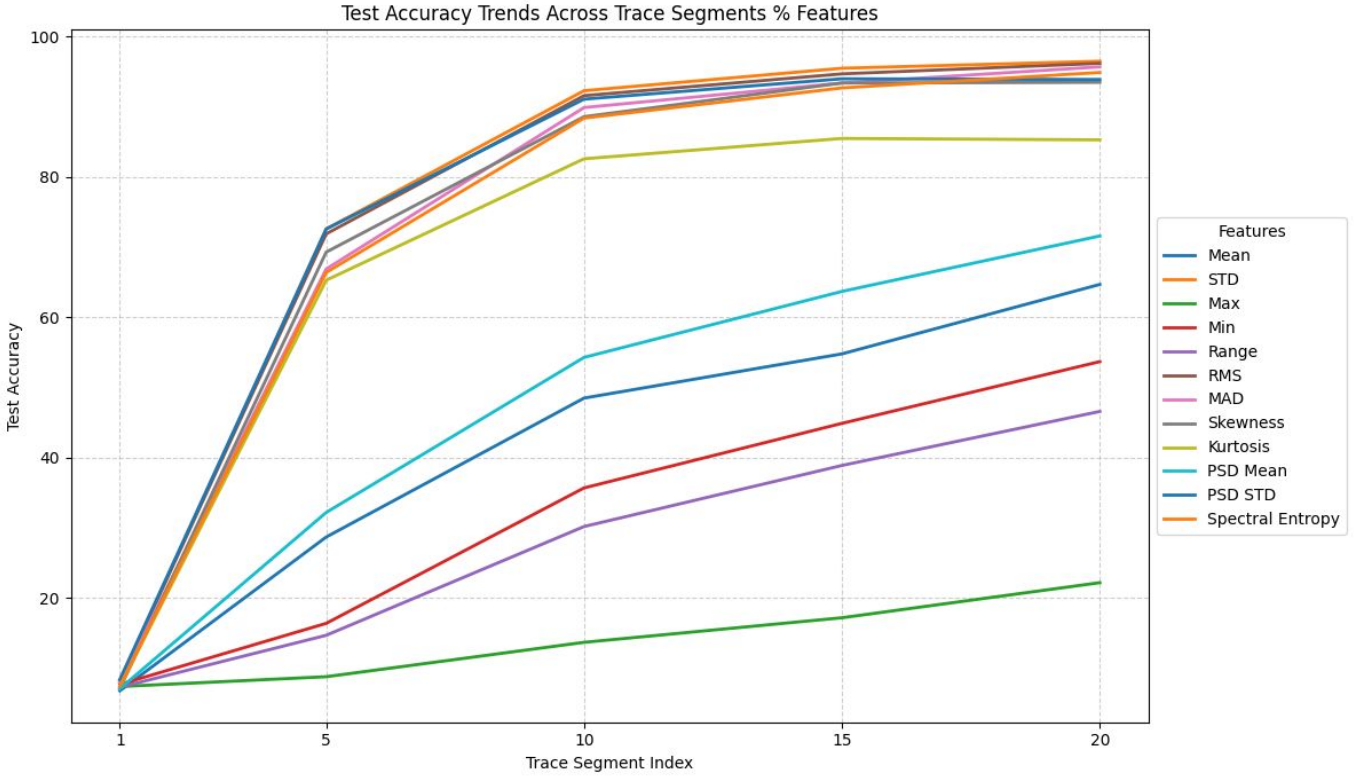
IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

LPSS

Statistical Features Considered in this Work

Feature	Description	Domain
mean	arithmetic mean	time
std	standard deviation	time
max	maximum value	time
min	minimum value	time
ptp	peak-to-peak (range)	time
rms	root mean square	time
mad	mean absolute deviation	time
skew	distribution's asymmetry	time
kurtosis	distribution's tailedness	time
psd_mean	power spectral density mean	frequency
psd_std	power spectral density std	frequency
spectral_entropy	Shannon's entropy of the PSD	frequency

Feature Ablation - Test Accuracy Function of Feature



The above graph was obtained by evaluating the test accuracy using only one feature at a time, using RF model and Multi-All config function of nb of segments S

Parameters Tuning - Features, Nb of segments, Dataset (HT coverage, accelerators)

Features

Among the top 6 features (STD, RMS, MAD, spectral-entropy, psd_std, skewness)
STD, RMS and MAD are likely correlated → MAD is more robust
spectral-entropy, psd_std, and skewness are orthogonal → less correlated
Suggested set of features: *MAD, spectral-entropy, psd_std, skewness*

Nb of segments

Goal: test accuracy → 98%, evaluation accuracy 92%

Dataset

Contains 4 accelerators, variety of HTs to cover most probable attacks

Feature Ablation - Test Accuracy Function of Feature

TEST ACCURACY EVOLUTION FUNCTION OF s PER FEATURE USING MULTI-ALL CONFIGURATION USING RF

Feature	$s = 1$	$s = 5$	$s = 10$	$s = 15$	$s = 20$
mean	6.8%	28.7%	48.5%	54.8%	64.7%
std	8.0%	72.6%	92.3%	95.5%	96.5%
max	7.4%	8.8%	13.7%	17.2%	22.2%
min	7.7%	16.4%	35.7%	44.9%	53.7%
ptp	7.3%	14.7%	30.2%	38.9%	46.6%
rms	8.3%	71.9%	91.6%	94.7%	96.6%
mad	8.0%	66.9%	89.9%	93.4%	95.7%
skew	7.1%	69.3%	88.6%	93.4%	93.5%
kurtosis	7.4%	65.3	82.6%	85.5%	85.5%
psd_mean	7.1%	32.2%	54.3%	63.7%	71.6%
psd_std	8.4%	72.6%	91.1%	94.0%	94.0%
spectral_entropy	7.5%	66.4%	88.4%	92.7%	94.4%

Feature Ablation - Test Accuracy Function of Feature

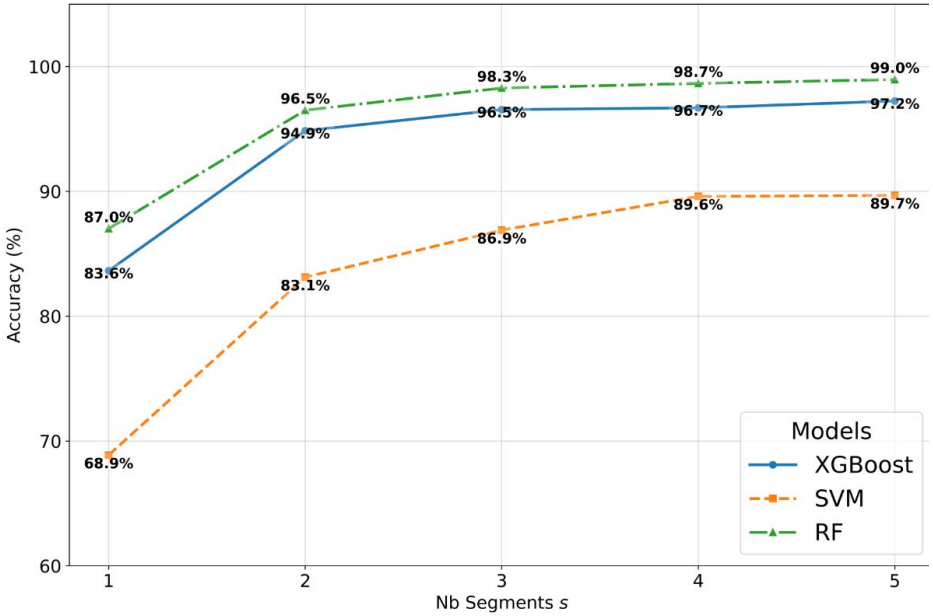
TEST ACCURACY EVOLUTION FUNCTION OF s PER FEATURE USING MULTI-ALL CONFIGURATION USING RF

Feature	$s = 1$	$s = 5$	$s = 10$	$s = 15$	$s = 20$	
mean	6.8%	28.7%	48.5%	54.8%	64.7%	
std	8.0%	72.6%	92.3%	95.5%	96.5%	
max	7.4%	8.8%	13.7%	17.2%	22.2%	
min	7.7%	16.4%	35.7%	44.9%	53.7%	
ptp	7.3%	14.7%	30.2%	38.9%	46.6%	
rms	8.3%	71.9%	91.6%	94.7%	96.6%	
mad	8.0%	66.9%	89.9%	93.4%	95.7%	1
skew	7.1%	69.3%	88.6%	93.4%	93.5%	4
kurtosis	7.4%	65.3	82.6%	85.5%	85.5%	
psd_mean	7.1%	32.2%	54.3%	63.7%	71.6%	
psd_std	8.4%	72.6%	91.1%	94.0%	94.0%	3
spectral_entropy	7.5%	66.4%	88.4%	92.7%	94.4%	2

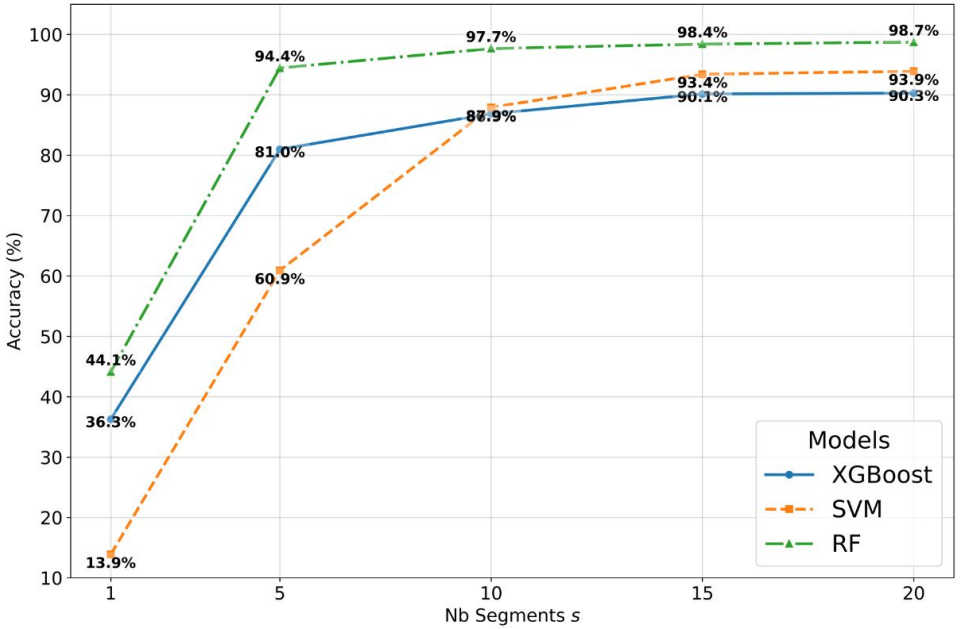
IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

LPSS
In Silicon, We Innovate

Accuracy Evolution Function of number of segments *S* - using 4 features



Binary-Diff configuration



Multi-All configuration

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

LPSS

* S segments was being incremented to reach the desired accuracy (+98%)

Evaluation using unseen HTs - Binary-class configuration

Benchmark	Accuracy	Precission	Recall	F1-Score
maxpool_1_pattern-comparator	83.1%	100%	83.1%	90.7%
maxpool_1_pattern-counter	92.6%	100%	92.6%	96.1%
reconfigurable_mvac_1	98.2%	100%	98.2%	99.1%
reconfigurable_mvac_2	98.7%	100%	98.7%	99.3%

* nb of segments S=3, using 4 features (MAD, spectral_entropy, psd_std, skew)

Comparison of the Proposed Work with Previous Works

Paper	HW for AI	Extracted Features	Number of Benchmarks	Detection Technique	Accuracy	Segments	Evaluation
[1]	❌	Raw data	12	CNN with residual blocks	90.48%	-	-
[2]	❌	PCA (5-dim.)	25	NB, DT, KNN, SVM	80%~99%	1	63.1%
[3]	❌	Statistical (min, max, mean, std.)	4	SVM, NB, RF	97%~100%	1	73.4%
[4]	❌	Raw data	1	Logistic regression	81%~95%	-	-
Our Work	✅	Statistical (4 features)	15	XGBoost, SVM, RF	98.2%~99.9%	2~3	99.7%

[1] A. Roy, "**Detection of Hardware Trojan using CNN with Residual Network**," in IEEE CICT, pp. 1–6, **2022**.
[2] S. Yang, "**Side-channel Analysis for Hardware Trojan Detection using Machine Learning**," in IEEE ITC India, pp. 1–6, **2021**.
[3] R. Gayatri, "**System Level Hardware Trojan Detection Using Side-Channel Power Analysis and Machine Learning**," in 5th ICCES Conference, **2020**.
[4] S. R. Zantout, "**Hardware Trojan Detection in FPGA through Side-Channel Power Analysis and Machine Learning**," in University of California, Irvine, **2018**.

Comparison of the Proposed Work with Previous Works

Paper	HW for AI	Extracted Features	Number of Benchmarks	Detection Technique	Accuracy	Segments	Evaluation
[1]	❌	Raw data	12	CNN with residual blocks	90.48%	-	-
[2]	❌	PCA (5-dim.)	25	NB, DT, KNN, SVM	80%~99%	1	63.1%
[3]	❌	Statistical (min, max, mean, std.)	4	SVM, NB, RF	97%~100%	1	73.4%
[4]	❌	Raw data	1	Logistic regression	81%~95%	-	-
Our Work	✅	Statistical (4 features)	15	XGBoost, SVM, RF	98.2%~99.9%	2~3	99.7%

AI accelerators remain underexplored → our work fills this gap

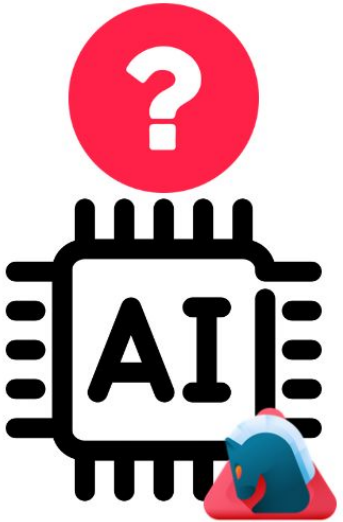
Our method outperforms traditional approaches on AI accelerators

Outline

1. Context and Background
2. Literature Review
3. **Proposed Approaches and Results**
   a. HT Presence Detection
   b. HT Impact Identification
4. Conclusion

Is HT Detection Enough for AI Accelerators?

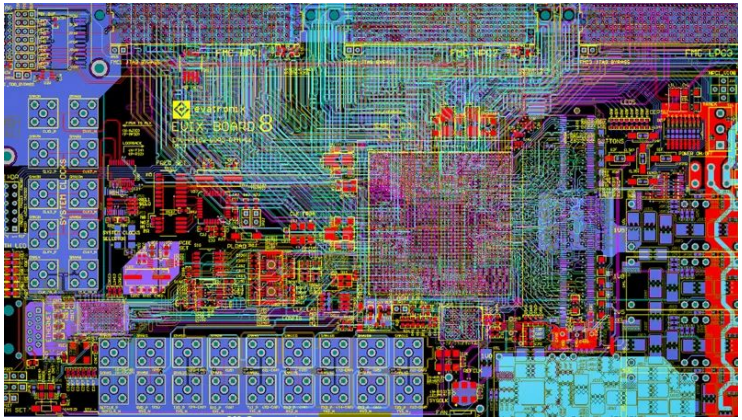
- HT detection in AI accelerators is now feasible, but detection alone is not enough
- Locating the HT or identifying impacted regions provides far more actionable security insight



Is HT Detection Enough for AI Accelerators?

- Existing methods focus on detection and suffer from limited coverage, precision, and scalability

- For large, complex accelerators, security audits need automated localization, manual analysis is impractical in fast-paced production



Proposed Approach*

BERT-Inspired HT Localization on FPGA AI Accelerators

AbdelSalam Baltagi[0000−1111−2222−3333], Yehya Nasser[1111−2222−3333−4444], and Amer Baghdadi[2222−−3333−4444−5555]

IMT Atlantique, Lab-STICC, UMR CNRS 6285, F-29238 Brest, France
firstname.surname@imt-atlantique.fr

Abstract. Outsourcing accelerates AI hardware development but exposes designs to hardware Trojan (HT) risks. The stealthy nature of HTs and increasing complexity of CNN accelerators limit the effectiveness of traditional detection and localization techniques. This paper presents a segmentation-driven power side-channel framework combined with a customized BERT-inspired model to localize HT-affected regions relative to a trusted baseline. By dividing traces into segments and analyzing them with transformer attention, the method enables targeted auditing and reduces manual inspection. Experiments on real power measurements with physical HTs achieve 92.1% localization accuracy on known HTs and 75.2% on unseen HTs. Contributions include a pre-processing pipeline for segmented trace analysis, a BERT-style architecture optimized for HT localization, and demonstrated generalization across multiple HT instances. The approach supports efficient, focused security audits in outsourced fabrication workflows.

Keywords: Hardware Trojan · AI Accelerator · Machine learning · BERT-inspired models · Power side-channel analysis.

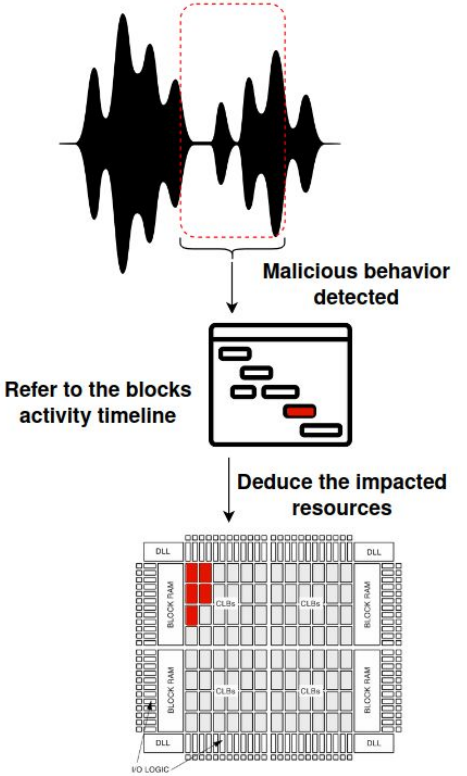
IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

LPSS
In Silicon, We Innovate

* This section presents our work published at the DASIP Workshop, co-located with the HiPEAC Conference: A. Baltagi, Y. Nasser, A. Baghdadi, “**BERT-Inspired HT Localization on FPGA AI Accelerators**, ” 2026

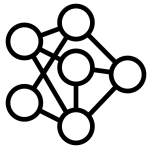
Proposed Approach*

- Detecting the presence of an HT alone is insufficient
- This is especially true for large and complex designs such as AI accelerators
- Our goal is to extract richer information from the inference power trace
- In particular, we aim to identify the blocks impacted by HT insertion



* This section presents our work published at the DASIP Workshop, co-located with the HiPEAC Conference: A. Baltagi, Y. Nasser, A. Baghdadi, "**BERT-Inspired HT Localization on FPGA AI Accelerators**, " 2026

HT Localization Recent Work



Graph-Based Netlist Learning

GNN-Based HT localization by transforming the netlist to a graph

[1] H. Su, “**Toward Precise and Explainable Hardware Trojan Localization at LUT Level,**” 2025.
[2] H. Zhang, “**B-HTRecognizer: Bitwise Hardware Trojan Localization Using Graph Attention Networks**,” 2025.
[3] R. Fan, “**An Efficient ML-based Hardware Trojan Localization Framework for RTL Security Analysis**,” 2024.
[4] Y. Li, “**HTrans: Transformer-Based Method for Hardware Trojan Detection and Localization**,” 2023.
[5] . Zhang, “**RLocHT: A Hardware Trojans Localization Method Utilizing Deep Learning at the Gate-Level**,” 2022.

HT Localization Recent Work

Graph-Based Netlist Learning

GNN-Based HT localization by transforming the netlist to a graph

Text-Based Netlist Analysis

Treating the netlist as text with NLP/ML models

[6] R. Yasaei, "**Golden Reference-Free Hardware Trojan Localization Using Graph Convolutional Network**," 2022.
[7] S. A. Islam, "**A Framework for Hardware Trojan Vulnerability Estimation and Localization in RTL Designs**," 2020.

HT Localization Recent Work

Graph-Based Netlist Learning

GNN-Based HT localization by transforming the netlist to a graph

Text-Based Netlist Analysis

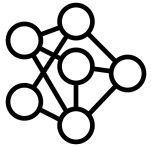
Treating the netlist as text with NLP/ML models

Side-Channel Temporal Analysis

Using side-channel information (time-series power/timing)

[8] M. Sabri, "**SAT-Based Integrated Hardware Trojan Detection and Localization Approach Through Path-Delay Analysis**," 2021.

HT Localization Recent Work



Graph-Based Netlist Learning

GNN-Based HT localization by transforming the netlist to a graph

Text-Based Netlist Analysis

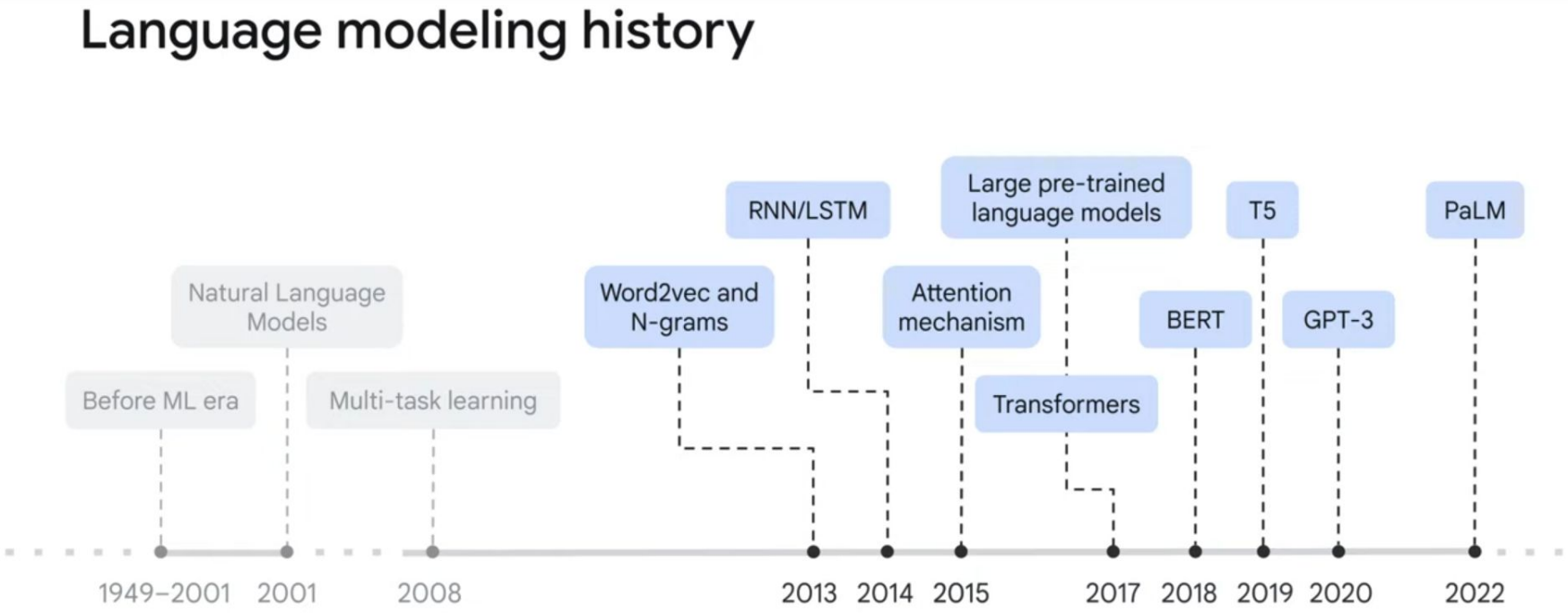
Treating the netlist as text with NLP/ML models

Side-Channel Temporal Analysis

Using side-channel information (time-series power/timing)

Power side-channel information is underexplored in current SOTA + Targeted cryptographic hardware

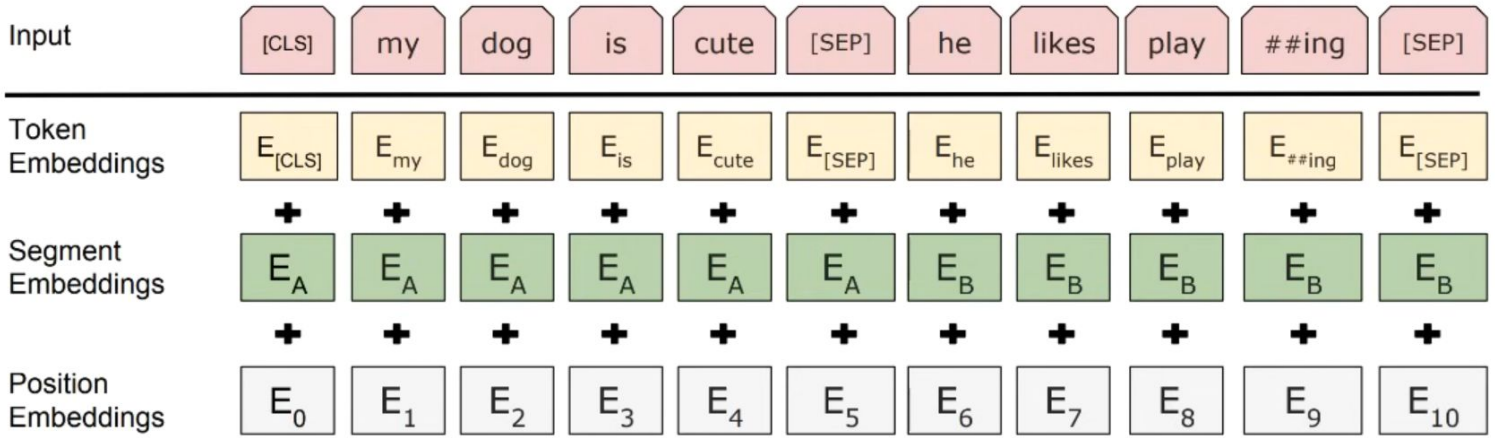
Background: Language Models



Background: Language Models

Bidirectional Encoder Representation from Transformers

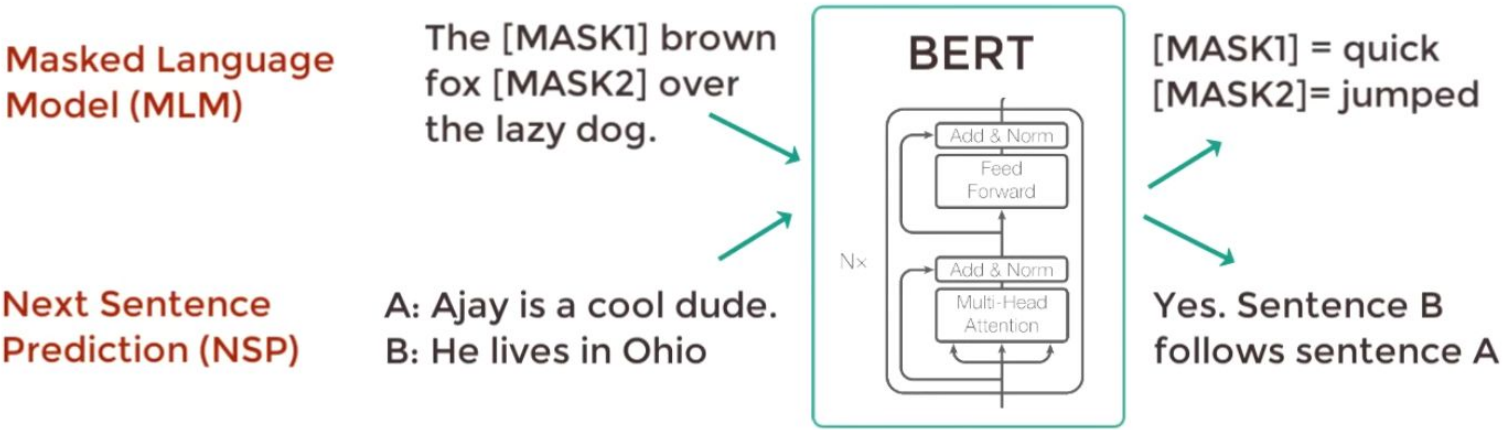
Pretraining (Pass 3)



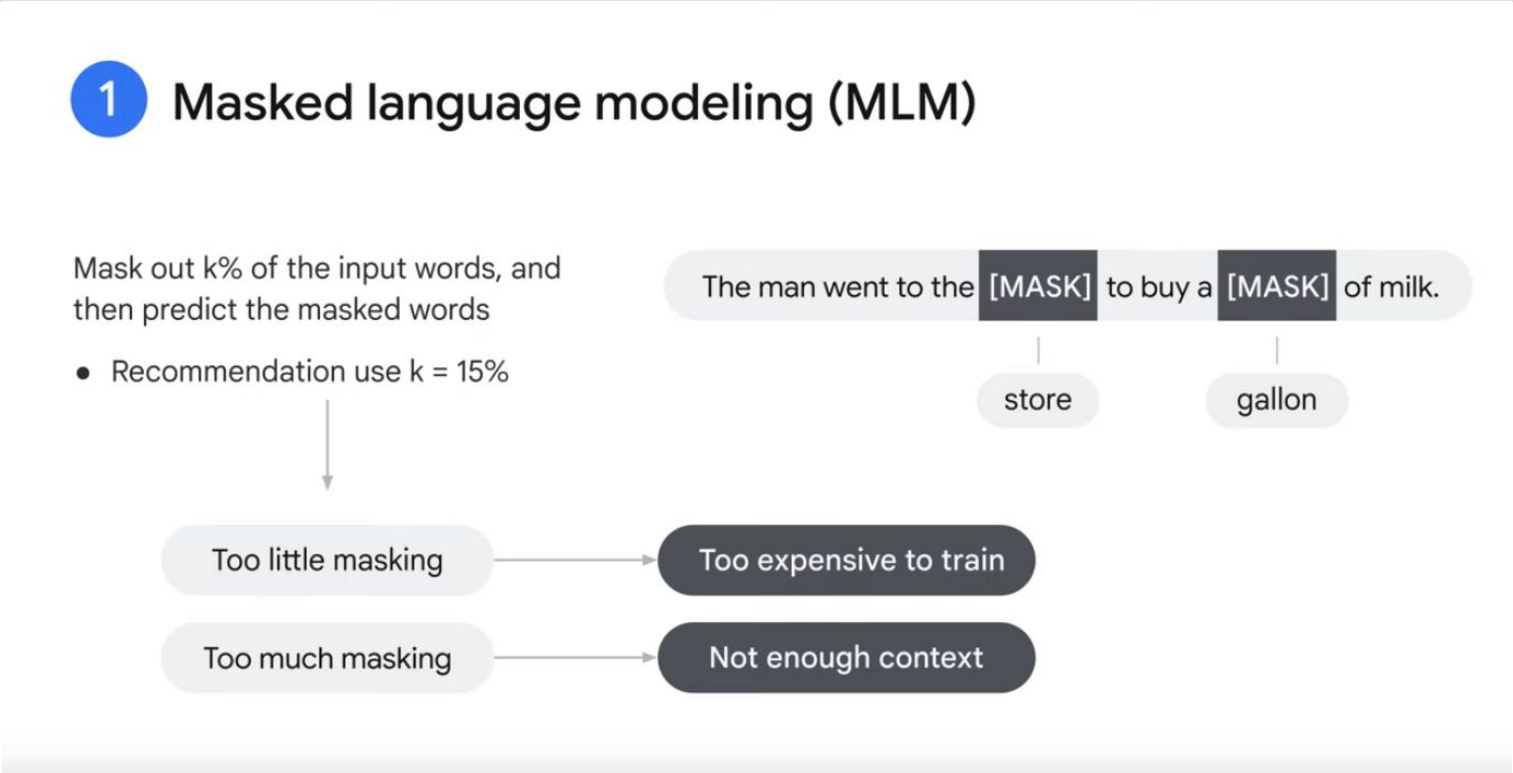
Background: Language Models

Bidirectional Encoder Representation from Transformers

Pretraining (Pass 1) : "What is language? What is context?"



Background: Language Models



Background: Language Models

2 Next sentence prediction (NPS)

Binary classification task

Learn the relationships between sentences and predict the next sentence given the first one.

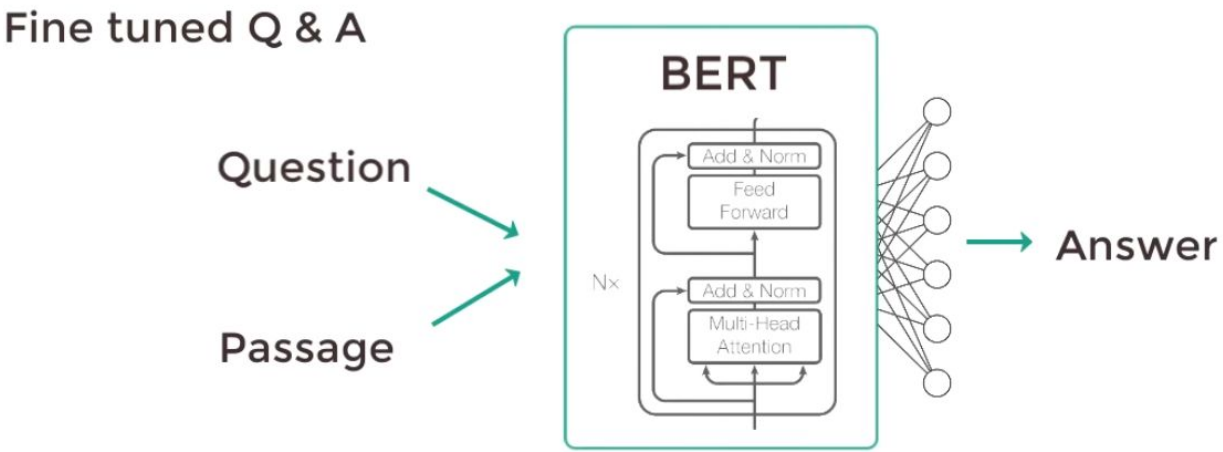
Sentence A	The man went to the store.
Sentence B	He bought a gallon of milk.
Label	IsNextSentence

Sentence A	The man went to the store.
Sentence B	Penguins are flightless.
Label	NotNextSentence

Background: Language Models

Bidirectional Encoder Representation from Transformers

Fine Tuning (Pass 1): "How to use language for specific task?"



How to Benefit from the Language Models Concept?

- The power trace is treated as a sentence, and its regions as words
- Each region is assigned a modification score reflecting resource utilization changes relative to the clean accelerator
- The model predicts these scores from feature representations of the power trace regions

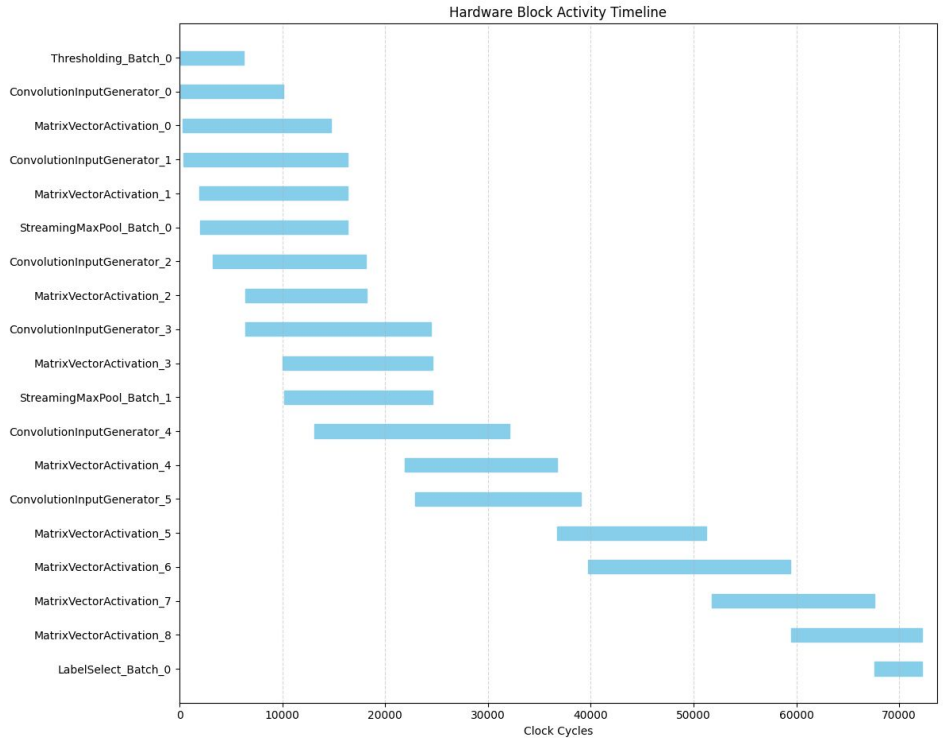
HT Impact Identification as a Sentence – Word Prediction Problem

- The hierarchical utilization report is extracted after implementation
- The report provides per-block resource usage (LUTs and FFs), as each block corresponds to a separate netlist

Instance	Module	Total LUTs	Logic LUTs	LUTRAMs	SRLs	FFs	RAMB36	RAMB18	DSP Blocks
finn_design_wrapper_top	(top)	33406	32112	656	638	67478	127	8	0
(finn_design_wrapper_top)	(top)	2	2	0	0	48	0	0	0
U_clocks	clocks	0	0	0	0	0	0	0	0
U_reg_aes	cw305_reg_aes	339	339	0	0	1726	0	0	0
(U_reg_aes)	cw305_reg_aes	335	335	0	0	1708	0	0	0
U_go_pulse	cdc_pulse	2	2	0	0	9	0	0	0
U_go_pulse_1	cdc_pulse_0	3	3	0	0	9	0	0	0
U_usb_reg_fe	cw305_usb_reg_fe	683	683	0	0	42	0	0	0
finn_design_i	finn_design	25289	23995	656	638	40926	127	8	0
ConvolutionInputGenerator_0	finn_design_ConvolutionInputGenerator_0_0	622	558	64	0	527	0	0	0
ConvolutionInputGenerator_1	finn_design_ConvolutionInputGenerator_1_0	811	635	176	0	626	0	0	0
ConvolutionInputGenerator_2	finn_design_ConvolutionInputGenerator_2_0	727	631	96	0	435	0	0	0
ConvolutionInputGenerator_3	finn_design_ConvolutionInputGenerator_3_0	825	681	144	0	436	0	0	0
ConvolutionInputGenerator_4	finn_design_ConvolutionInputGenerator_4_0	710	630	80	0	382	0	0	0
ConvolutionInputGenerator_5	finn_design_ConvolutionInputGenerator_5_0	564	468	96	0	317	0	0	0
LabelSelect_Batch_0	finn_design_LabelSelect_Batch_0_0	56	56	0	0	62	0	0	0

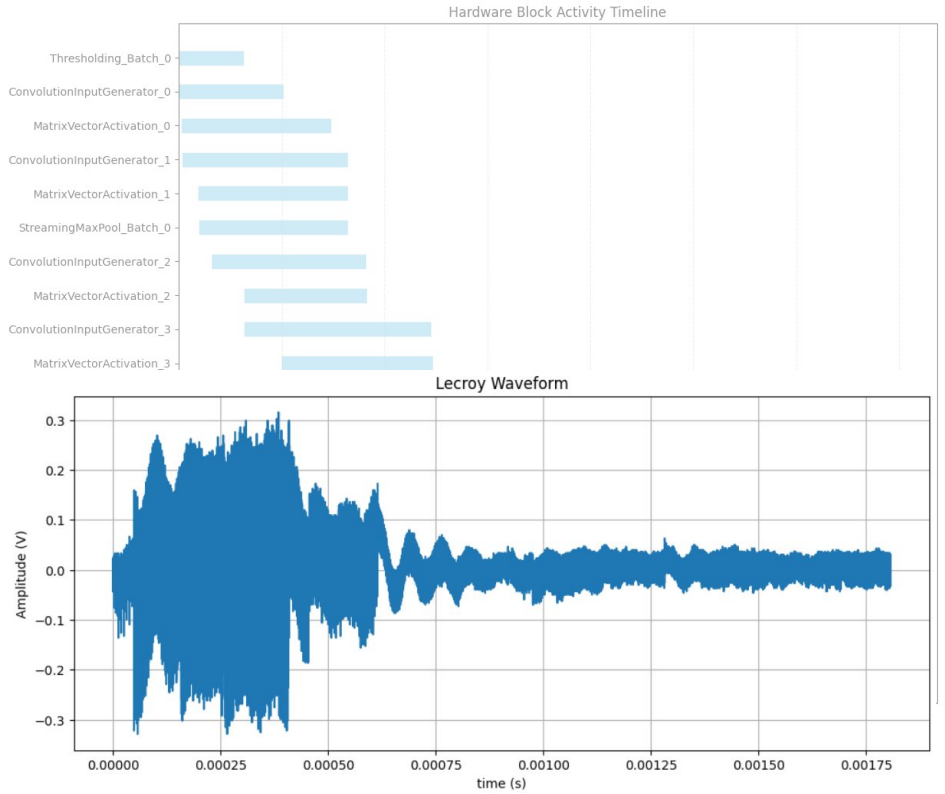
HT Impact Identification as a Sentence – Word Prediction Problem

- Each block is assigned a modification score equal to the difference between infected and clean resource utilization
- A zero modification score across all blocks indicates a clean accelerator



HT Impact Identification as a Sentence – Word Prediction Problem

- Each block is assigned a modification score equal to the difference between infected and clean resource utilization
- A zero modification score across all blocks indicates a clean accelerator
- Since blocks operate in parallel, how can 2D block-level scores be mapped to a 1D power trace?

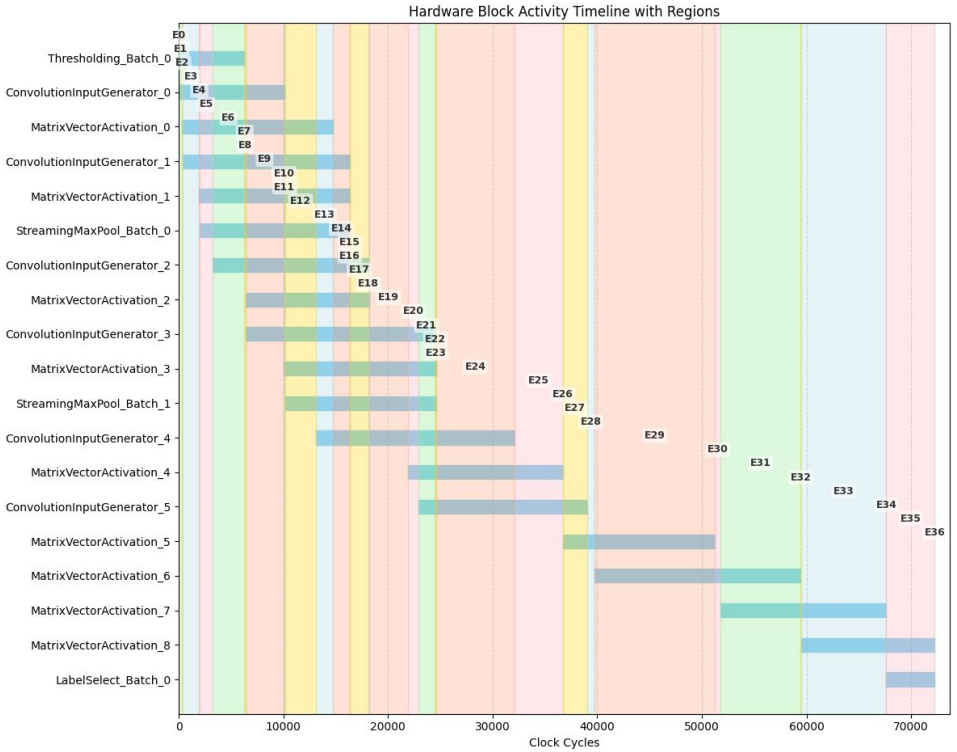


HT Impact Identification as a Sentence – Word Prediction Problem

- The power trace is segmented according to the block activity timeline into **Regions.** (Blocks → Regions)
- Each region corresponds to a unique combination of active blocks
- First, each block is assigned a modification score as following:

$$block\_score_i = (LUT_i + FF_i) - (LUT_{clean} + FF_{clean})$$

$$i \in \{0, 1, \ldots, N_{blocks}-1\}$$



HT Impact Identification as a Sentence – Word Prediction Problem

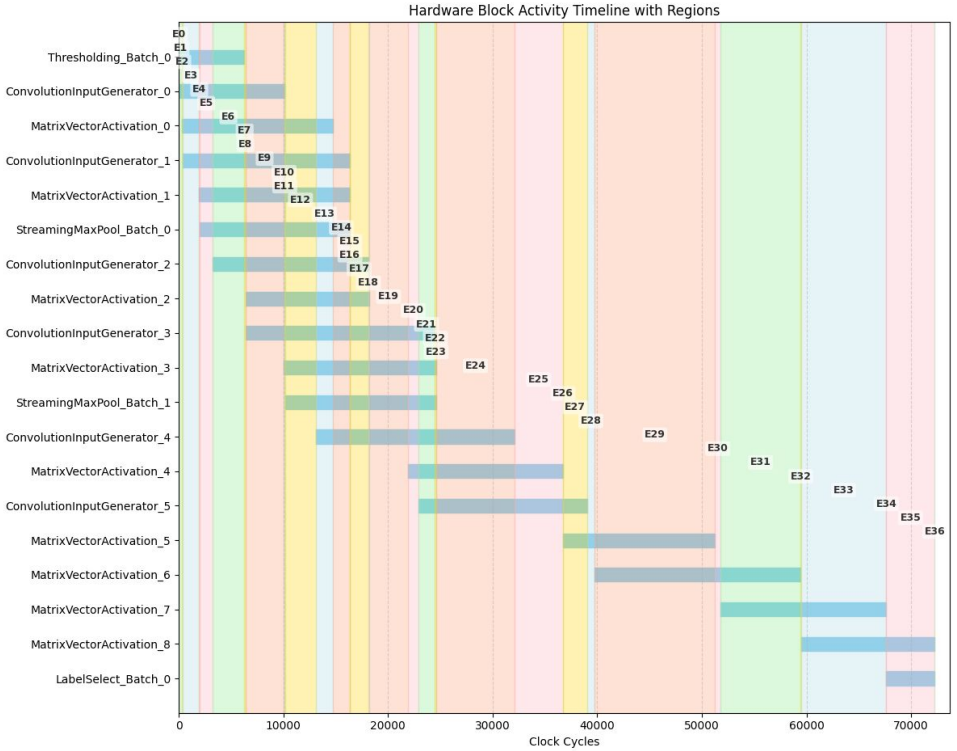
- Each region is assigned the sum of modification scores of its active blocks

$$region\_score_j = \sum block\_score_i$$

$$i \in \{0, 1, \ldots, N_{blocks}\text{-}1\}$$
$$j \in \{0, 1, \ldots, N_{regions}\text{-}1\}$$

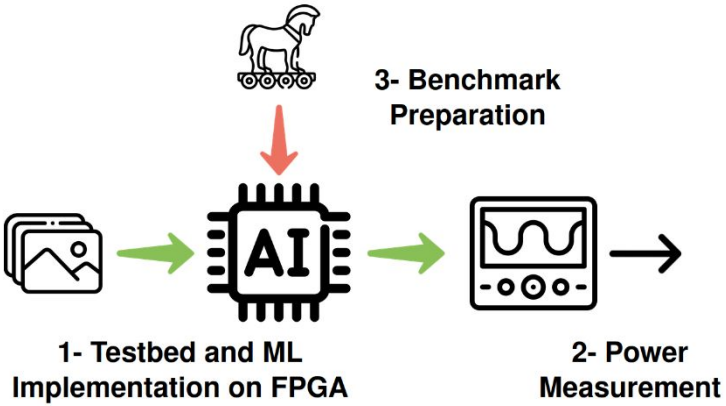
- For simplicity, the scores are transformed to a binary scores

$$mod\_score_j = \begin{cases} 0 & \text{if } region\_score_j < thresh. \\ 1 & \text{if } region\_score_j \geq thresh. \end{cases}$$

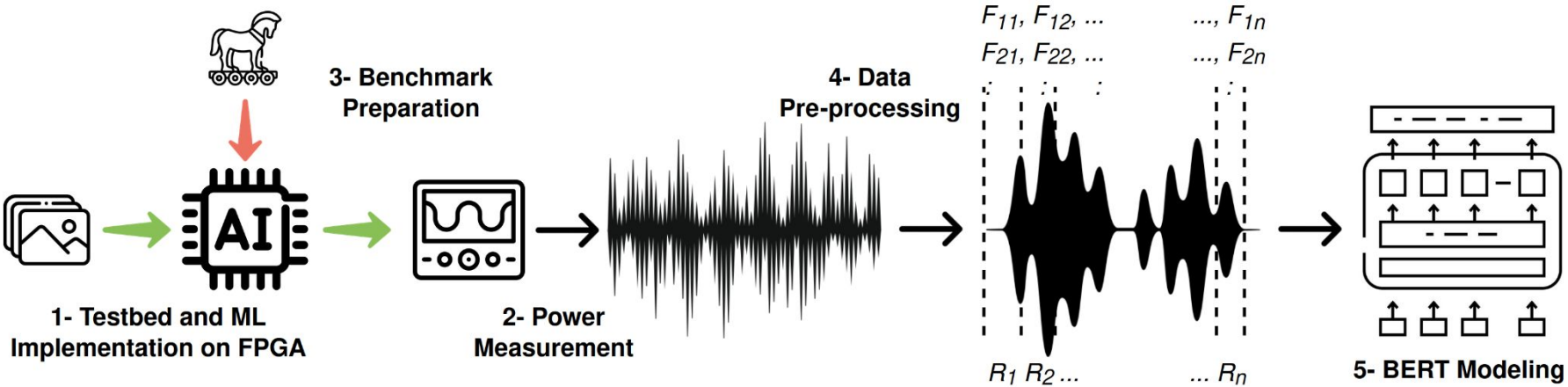


* the *thresh*. Is an adjustable threshold, example 1, 2, 3 …

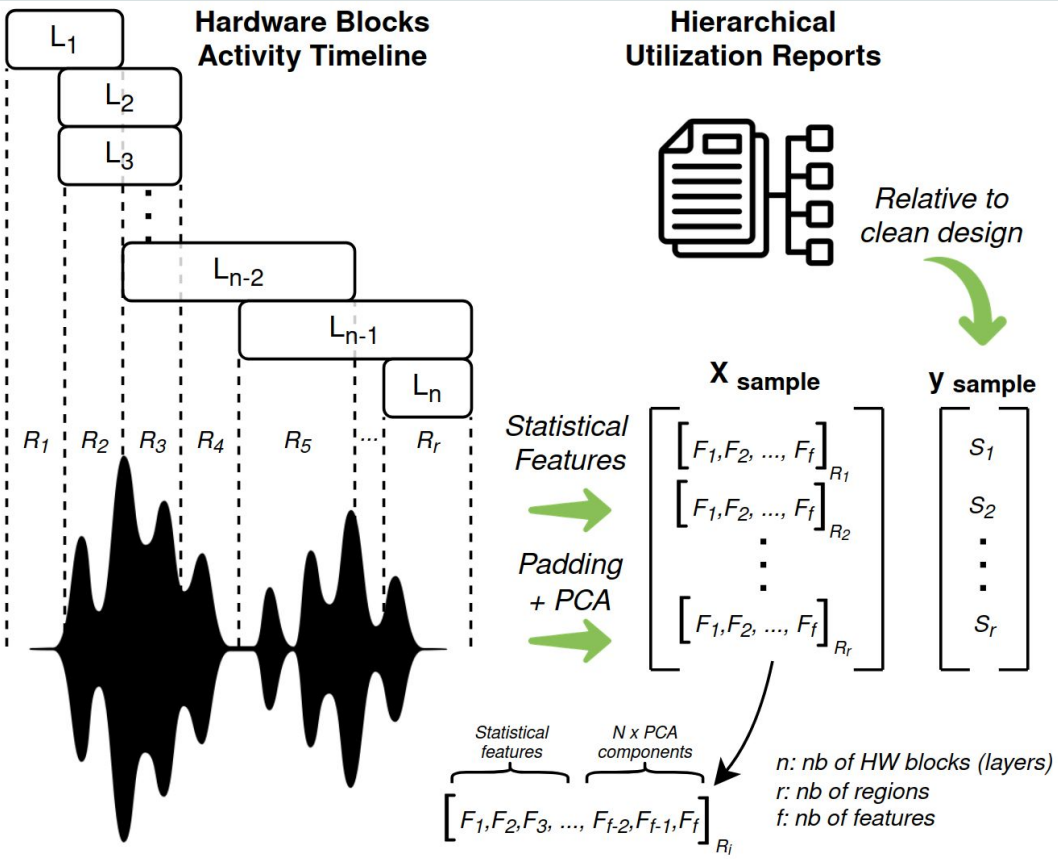
Main Workflow



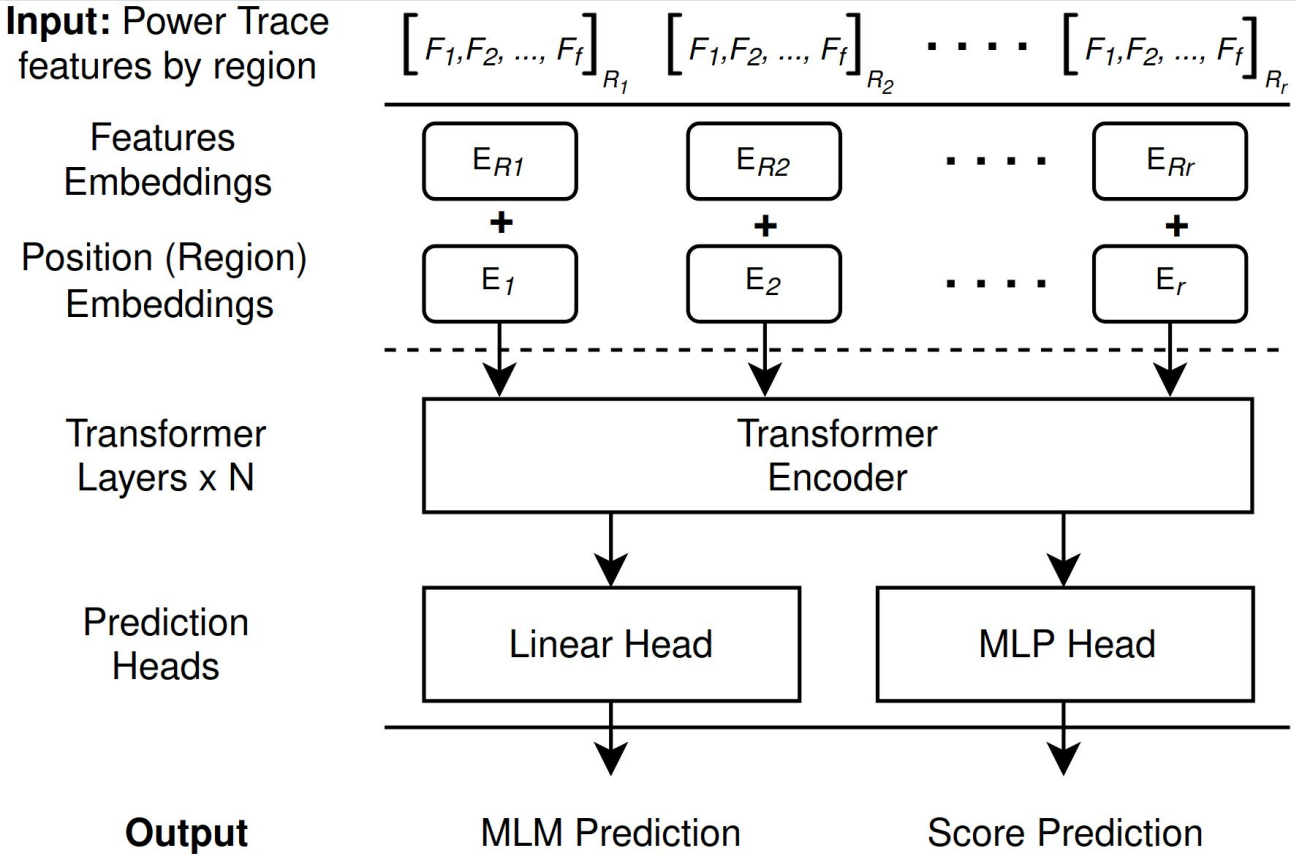
Main Workflow



Main Workflow - Step 4: Data Pre-processing



Main Workflow - Step 5: BERT-Style Modeling



Identification Accuracy Without/With MLM Head (*thresh.* = 2) - Seen HTs

Model Architecture	Test Accuracy
Transformer - **No MLM Head** MLP Classification Head only	- Class Task: Acc = 84.6%
Transformer - **MLM Head** + MLP Classification Head	- MLM Task: R^2 = 0.96 - Class Task: Acc = 92.1%

Identification Accuracy Without/With MLM Head (*thresh.* = 2) - Seen HTs

Model Architecture	Test Accuracy
Transformer - **No MLM Head** MLP Classification Head only	- Class Task: Acc = 84.6%
Transformer - **MLM Head** + MLP Classification Head	- MLM Task: R^2 = 0.96 - Class Task: Acc = 92.1%

Per-HT Accuracy on the Evaluation Dataset using the MLM head - Unseen HTs*

HT Name	$thresh = 1$	$thresh = 2$	$thresh = 3$
ext_comb_2	41.8%	36.2%	43.7%
ext_seq_4	60.4%	60.2%	59.4%
relu_fc7	87.8%	91.8%	85.8%
ht_shft_10_MsCls	89.0%	94.4%	91.4%
ht_shft_5_MsCls	72.5%	72.9%	71.0%
ht_shft_1_MsCls	71.2%	72.0%	70.1%
ht_MxTgMs	91.9%	94.0%	86.4%
ht_SwSTMs	98.2%	99.7%	92.1%
maxpool_0_always-on	65.9%	65.0%	65.2%
maxpool_0_pattern-comparator	66.6%	65.0%	64.8%
maxpool_0_pattern-counter	59.3%	57.0%	57.0%
reconfigurable_mvac_0	59.3%	55.4%	61.5%
geometric_trojan	70.6%	72.8%	70.3%
clock_glitch_always-on	88.7%	98.8%	95.6%
clock_glitch_pattern-recognition	87.3%	92.9%	97.0%
Average	70.5%	75.2%	74.9%

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

LPSS
In Silicon, We Innovate

* Unseen HTs are unknown designs (out-of-distribution)

Outline

1. Context and Background
2. Literature Review
3. Proposed Approaches and Results
4. **Conclusion**

Main Achievements Summary (1)

- Proposed a practical ML-based framework for detecting and identifying hardware Trojans in AI accelerators, achieving up to 99% accuracy.
- Trained classifiers on power traces from multiple accelerators with several HT insertions to detect HT presence and key traits (trigger, payload, always-on).
- Segmented long inference power traces and extracted simple time- and frequency-domain statistical features, outperforming more complex higher-order features.
- Designed for real-world integration, enabling post-detection actions such as discarding, restricting, or auditing compromised accelerators.

Main Achievements Summary (2)

- AI accelerators increase HT risks, making detection alone insufficient for long and highly dynamic power traces
- We propose a novel PSCA-based approach to localize HT-impacted regions in AI accelerators
- A BERT-style localizer models power traces as sentences, capturing contextual dependencies and improving accuracy even with minimal HTs
- This work enables region-level impact identification and supports variable-length traces, paving the way toward finer block-level localization

Lab-STICC
UMR 6285

THANK YOU

Yehya NASSER
Maitre de Conference / Associate Professor
yehya.nasser@imt-atlantique.fr