Detection and Classification of Malicious Bitstreams for FPGAs in Cloud Computing

Jayeeta Chaudhuri
Department of Electrical and Computer Engineering
Duke University

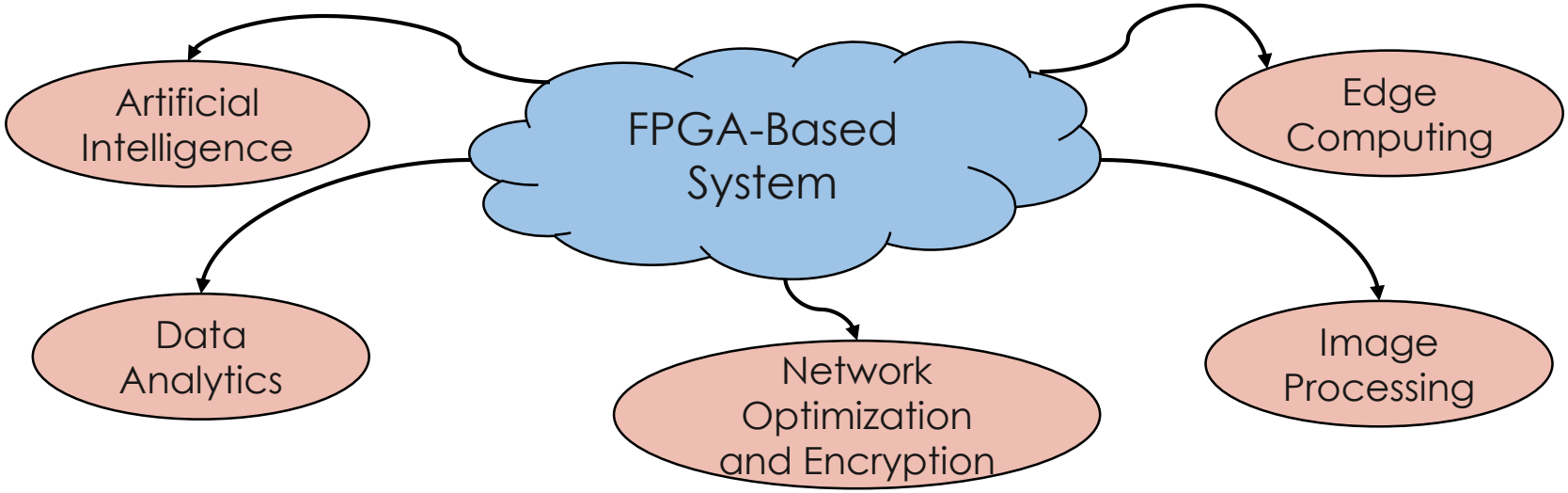
Krishnendu Chakrabarty
School of Electrical, Computer and Energy Engineering
Arizona State University

Outline

- Introduction
  - ✓ Overview of FPGAs in Cloud Computing
  - ✓ Security Threats to FPGAs
- Current Malicious Bitstream Detection Systems
- Proposed CNN-Based Detection
- Criticality Analysis of Malicious Bitstreams
- Conclusion

FPGAs Deployed in the Cloud

- FPGAs in cloud computing scenarios
  - ✓Low-cost, high-performance computing
  - ✓Customized accelerators – compute-intensive workloads



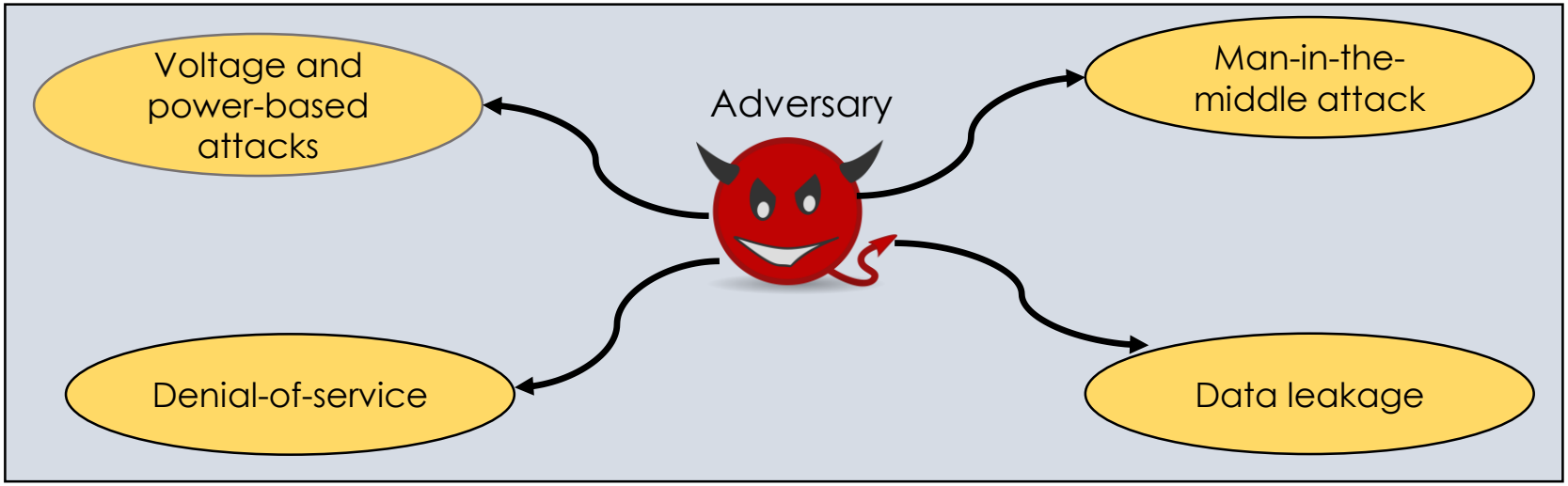
Multi-Tenant Scenario

- **FRIES**
  - ✓**F**lexibility
  - ✓**R**eliability
  - ✓**I**mproved Performance
  - ✓**E**fficiency
  - ✓**S**calability

FPGA-as-a-service:
Users → customized applications → deployed on cloud FPGA

Security and Trust in FPGA-Based Computing

- Multiple tenants – sharing same hardware
  - ✓ Adversary → FPGA configuration → harm tenant modules



Countermeasures

AWS Design Rule Check (DRC)

✓ Verify area and power constraints

✓ Timing analysis

✗ Loop-free RO not detected

FPGADefender [1]

✓ FPGA bitstreams → netlist graphs

✓ Both combinational and self-clocked ROs detected

✗ Requires reverse-engineering (RE) → significant time overhead

[1] T. M. La et al., "FPGADefender: Malicious self-oscillator scanning for Xilinx UltraScale + FPGAs," ACM TRETS, 2020.

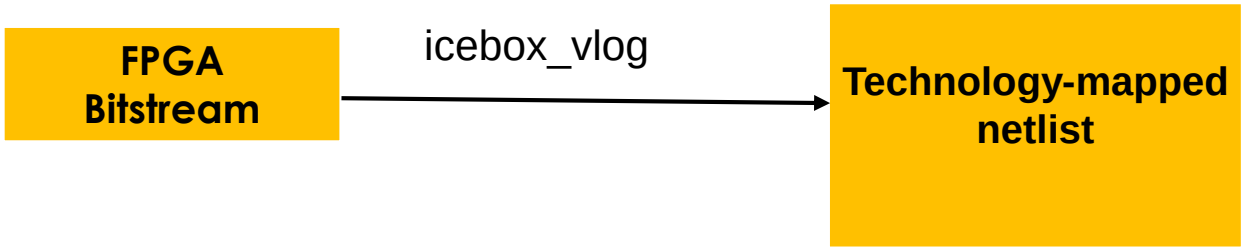
Reverse Engineering-Based Techniques

✓ Identify malicious signatures in FPGA bitstreams

✓ User-friendly *icebox_vlog* tool for analysis

✗ Not suitable for large designs

✗ Netlist creation → adds to time overhead



Comparison with Prior Work

Characteristics	[1]	[2]	[3]	**Our work**
RE used	✔	✔	✔	✗
Detects self-clocked RO	✗	✔	✗	✔
Detects conditional RO	✗	✗	✗	✔
Performs criticality analysis	✗	✗	✗	✔

[1] Dennis Gnad et al., "Checking for Electrical Level Security Threats in Bitstreams for Multi-tenant FPGAs", FPT, 2018
[2] T. M. La et al., "FPGADefender: Malicious self-oscillator scanning for Xilinx UltraScale + FPGAs," ACM TRETS, 2020
[3] Hassan Nassar et al., "LoopBreaker: Disabling Interconnects to Mitigate Voltage-Based Attacks in Multi-Tenant FPGAs", ICCAD, 2021

Motivation for New Research

- Prior methods mostly utilize RE-based techniques
  - ✓ Does not scale with complex designs
  - ✓ Time-consuming, resource intensive
- Need for machine learning (ML) -based techniques for malicious bitstream detection
- Power-hungry RO variants that evade AWS DRC
- Need for criticality analysis of FPGA bitstreams

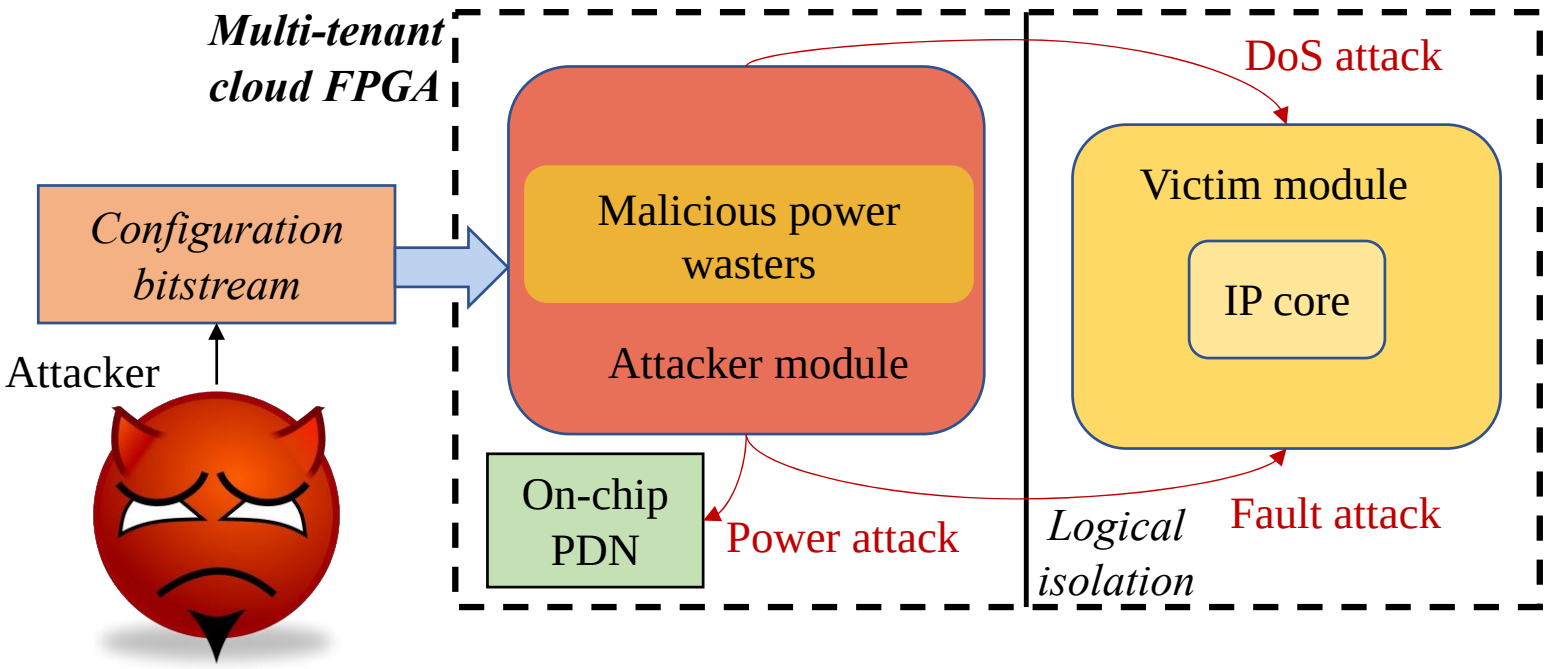
Contributions

1. Generation of new RO variants
   - Loop-free ROs: a rising threat to cloud FPGAs

2. CNN-based classification framework
   - Feature extraction from bitstream itself
   - Learn RO-based signatures through static analysis
   - Evaluation on diverse set of real-world bitstreams

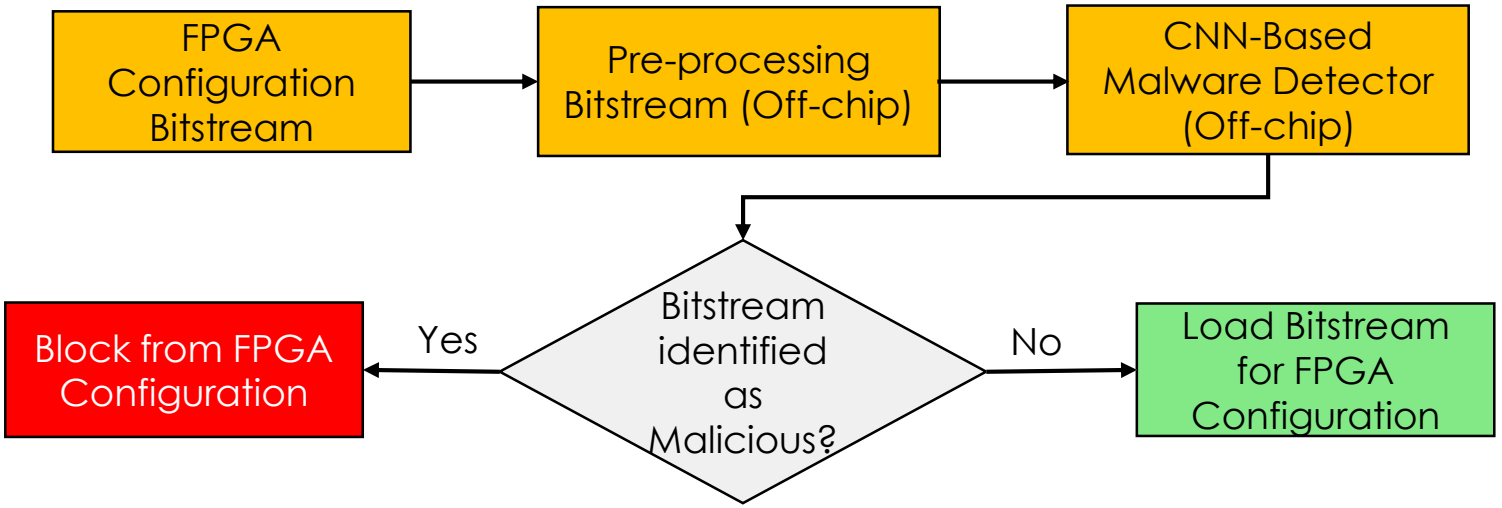
3. Criticality classification of FPGA bitstreams
   - Based on frequency-domain representation

Threat Model

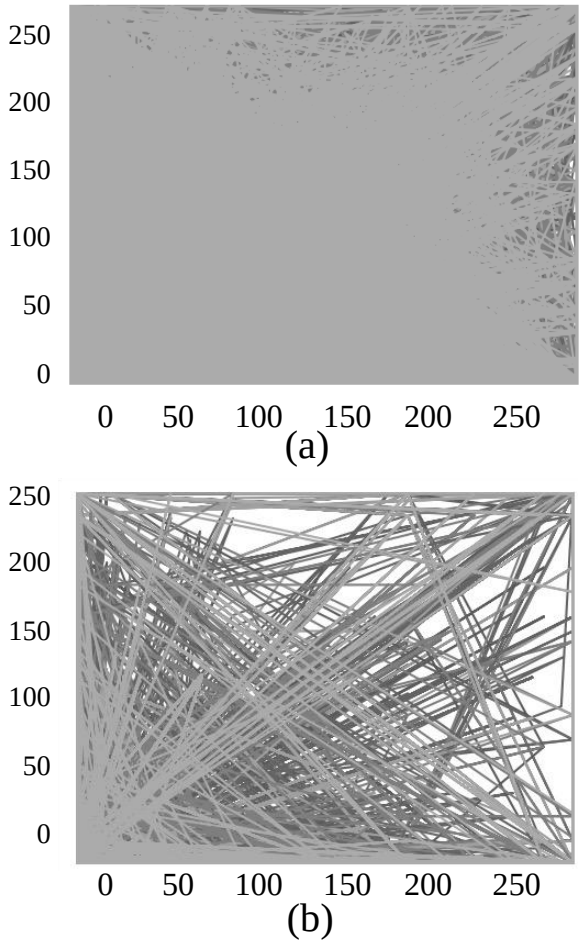


- Attacker and victim modules – logically isolated
- Attacker → malicious bitstream → FPGA → DoS

Bitstream Detection Pipeline



Mapping Bitstreams to Data-Series

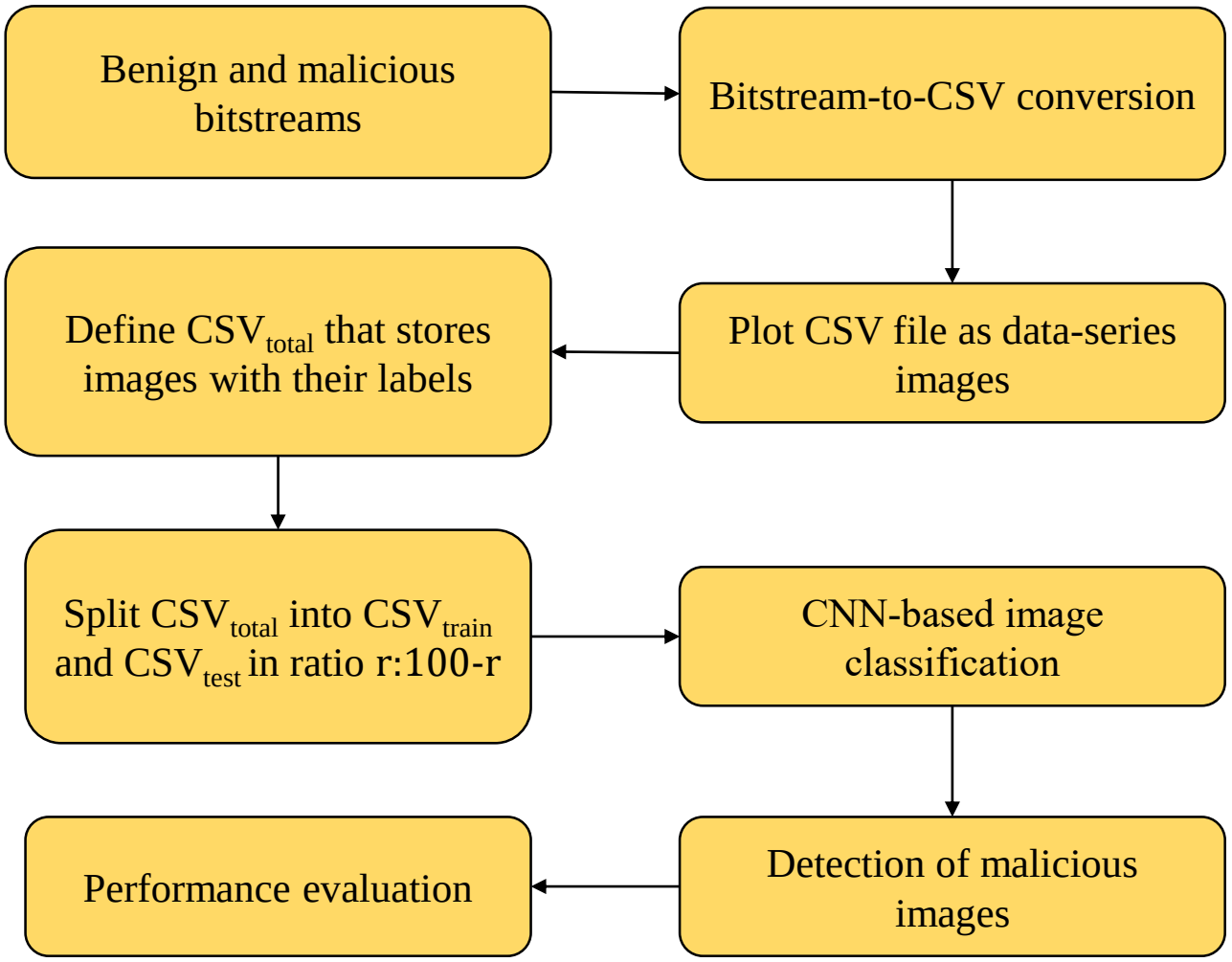


Apply **image augmentation**

- Expand training dataset with realistic examples from existing training data
- Enhance model performance to extract meaningful features

Images corresponding to: (a) benign bitstream; (b) malicious bitstream

Model Training and Evaluation



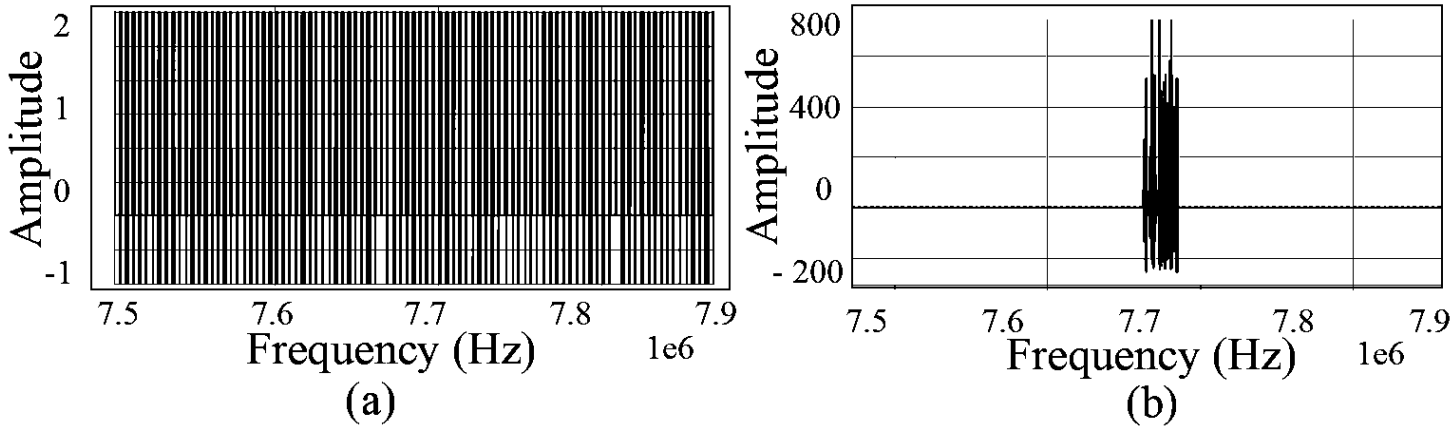
Need for Criticality Analysis of Bitstreams

- Presence of RO-based signatures – Bitstreams inappropriately blocked from FPGA configuration

- ML-based criticality analysis framework
  - Detect and block RO-based Trojans
  - Evaluated for diverse set of malicious bitstreams

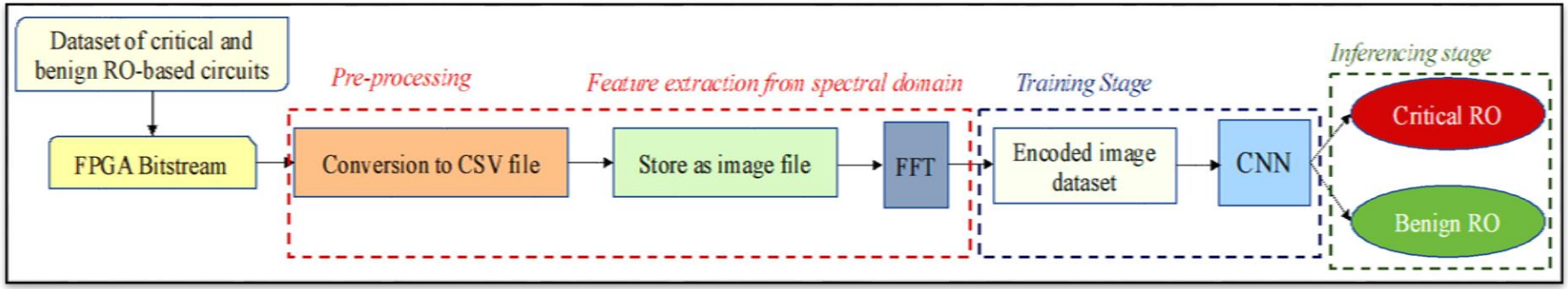
Feature Extraction in the Spectral Domain

- Fast Fourier Transform (FFT)
  - ✓ Time complexity of $O(n\ logn)$ - suitable for large bitstreams
  - ✓ Exploratory analysis of FPGA bitstreams



FFT-encoded images: (a) Benign RO (b) Critical RO.

Criticality Analysis Framework

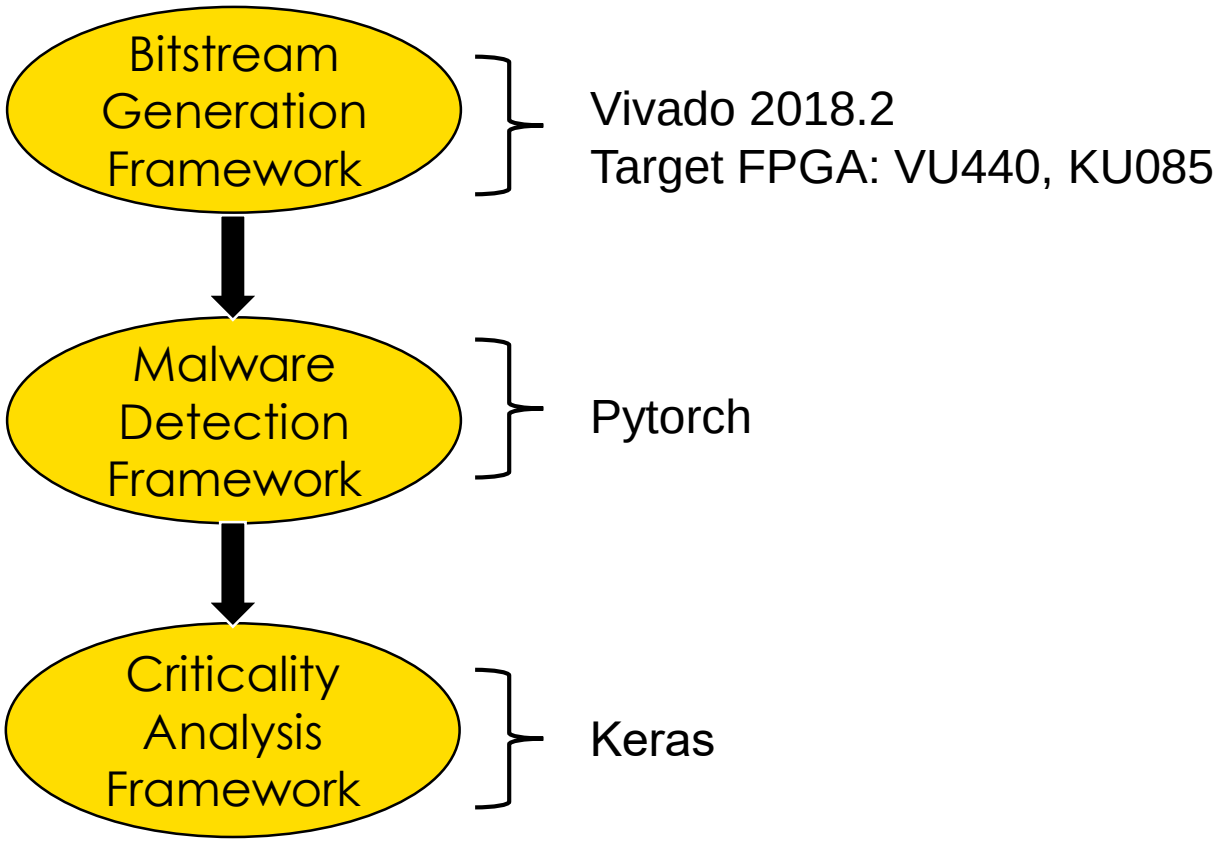


1) FFT-Based Feature Extraction: Extract frequency-domain features from input bitstream

2) CNN Training: Utilize frequency-domain features as model input

3) Evaluation: Evaluation methods such as classification accuracy (A_c), F1-score

Experimental Flow



Malware Detection Framework

- Neural network model: CNN
  - ✓ Four convolutional, four max pooling, and four linear layers
  - ✓ Input: grayscale image (classification)

- Dataset generation:
  - ✓ 95 benign and 80 malicious image files
  - ✓ After image augmentation: 250 image files

Selection of Model Hyperparameters

Hyperparameters	Best values
Learning rate	7.5e-5
Optimizer	Adam
Loss function	Cross Entropy
Number of training epochs	300
Dropout probability	0.25

Exploring *flip* Techniques

Technique	Training acc. (%)	Test acc. (%)
$flip_{lr}$	93.9	95.7
$\boldsymbol{flip_{ud}}$	**99.2**	**96.4**
$flip\ (axis(1,2))$	90.8	87.4

Choose appropriate image augmentation technique: $flip_{ud}$

Evaluation Results

Performance Metrics:

1. TPR_{mal}: Percentage of malicious bitstreams correctly classified as malicious

II. FPR_{mal}: Percentage of benign bitstreams incorrectly classified as malicious

Metrics (%) \ FPGA Board	VU440	KU085
TPR_{mal}	**97.08**	**95.83**
FPR_{mal}	4.29	7.5

VU440 –
- Training accuracy: **99.2%**
- Test accuracy: **96.4%**

KU085 –
- Training accuracy: **98.4%**
- Test accuracy: **95.7%**

Time Overhead

- Conversion of user-input bitstream to data-series:
  - Experimentation Platform - 2.4 GHZ Intel Xeon Gold 5115 CPU with 768 GB of RAM
  - Less than 4 minutes of CPU time

- CNN inferencing
  - Experimentation Platform - NVIDIA GeForce GTX 1080 GPU
  - Takes around 0.03s

Criticality Classification

Decision	Power-wasting RO	Cond. RO	Latched RO	Self-clocked RO	TRNG
Critical	9	6	4	6	1
Benign	0	0	2	1	19
A_c	100	100	66.67	85.7	95
Average A_c	**89.47**				

Classification accuracy (A_c): Ratio of the number of correct predictions to the total number of predictions.

- Convert FPGA bitstream-generated images to spectral domain
- Detect critical RO-based Trojans

Conclusion

- Demonstration of an efficient CNN-based malicious bitstream detection framework
- Accurate criticality classification of RO-based circuits
- Easily extended to other FPGA families, with minimum modifications