Hardware Trojan Detection and High-Precision Localization in NoC-based MPSoC using Machine Learning

Haoyu Wang, Basel Halak

Advancing Cyber Security Group
School of Electronic and Computer Science
University of Southampton, UK

Haoyu.wang@soton.ac.uk

Contents

Motivation: NoC (mesh based)

Scalable

Flexible

Shared-Resources

Extendable

High-Parallel

……

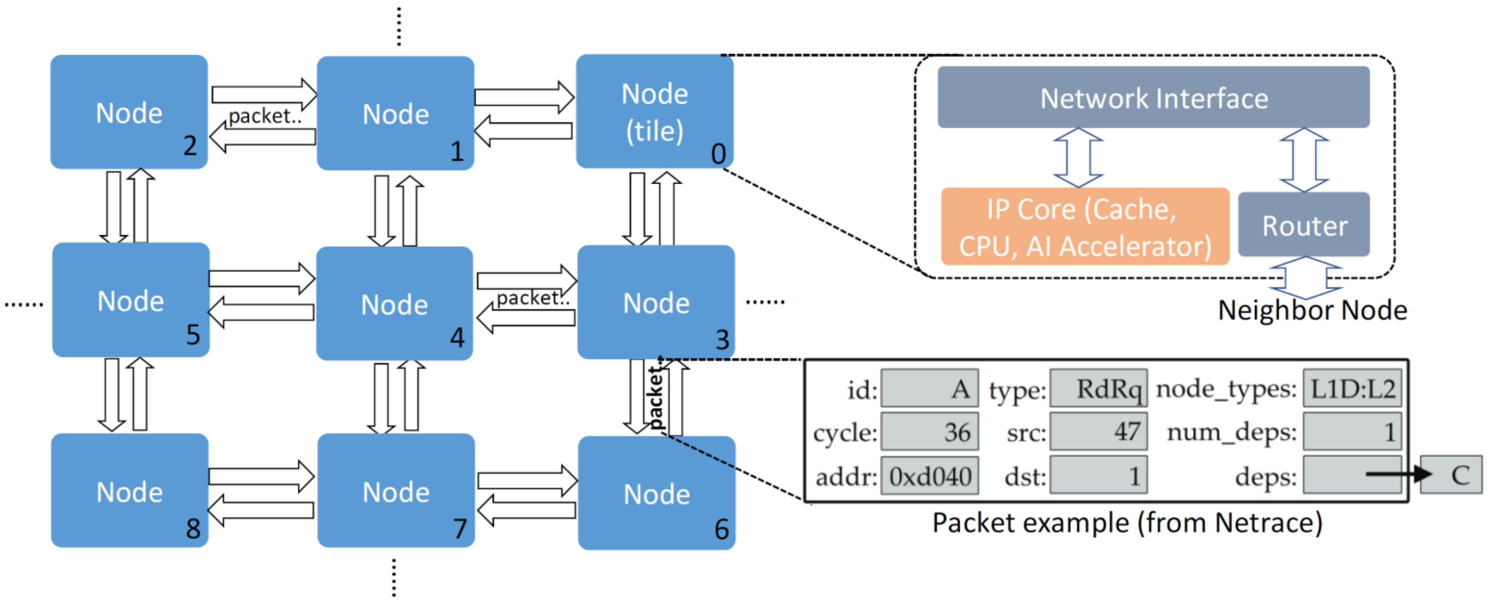


Fig1. Typical NoC architecture

- NoC is one of the better solutions for **Multi-Processer SoC (MPSoC)** design (Data center chips, AI chips, cloud computing infrastructures…)
  - More functionalities, higher performance, and a shorter R&D period
- Needs more outsourced 3PIPs (untrusted?) and licensed 3PEDAs (untrusted?)
  - The increasing number of security attacks that undermine the NoC

Motivation: Attacks on NoC

- Security attacks on NoC [1]: Eavesdropping, Spoofing, **Denial-of-Services**, Buffer Overflow, Side Channel
- DoS: **Tampering Attack**

Node — Normal node

Victim Node — Victim node

Malicious Node — Malicious node (HT injected)

Tampered Packets — Attacked/Tampered packets transferring

Attacking example0: Routing path tampering attack

Attacking example1: Source node initiates tampering attack

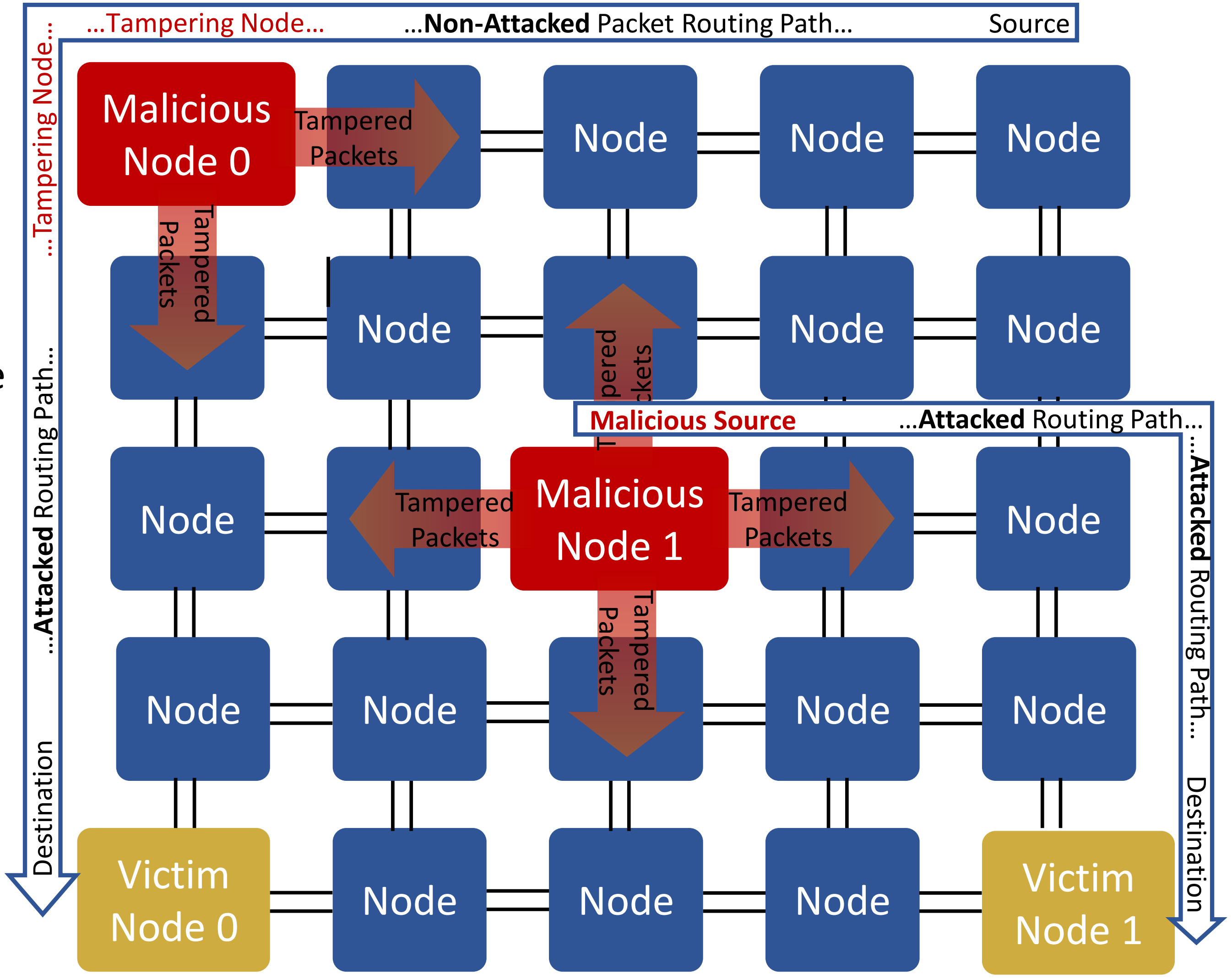


Fig2. Packet tampering attack examples

Threat model (adversary capability)

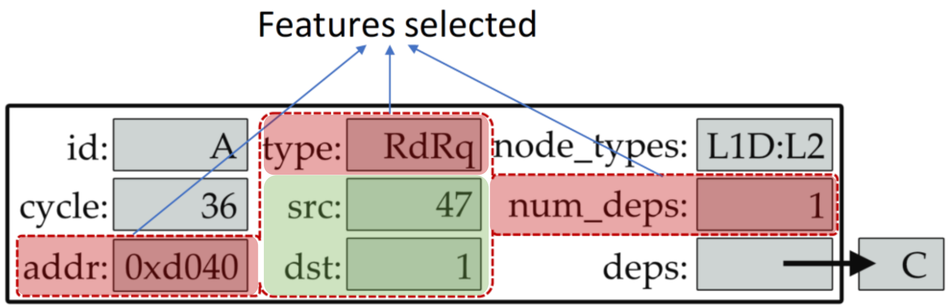


Fig3. Packet example

- **Tampered packet data:** memory address, type of the instruction, number of dependencies

- **Safe packet data:** source node ID, destination node ID
- Reason: leading to other unwanted attacks such as traffic diversion[2], route looping[2], and flooding

Proposed framework: Overview & DCI

- Proposed Framework: attack detection by DCI, HT localization by DSCT
- Tampering attack detection: Dynamic Confidence Interval (DCI) and ML

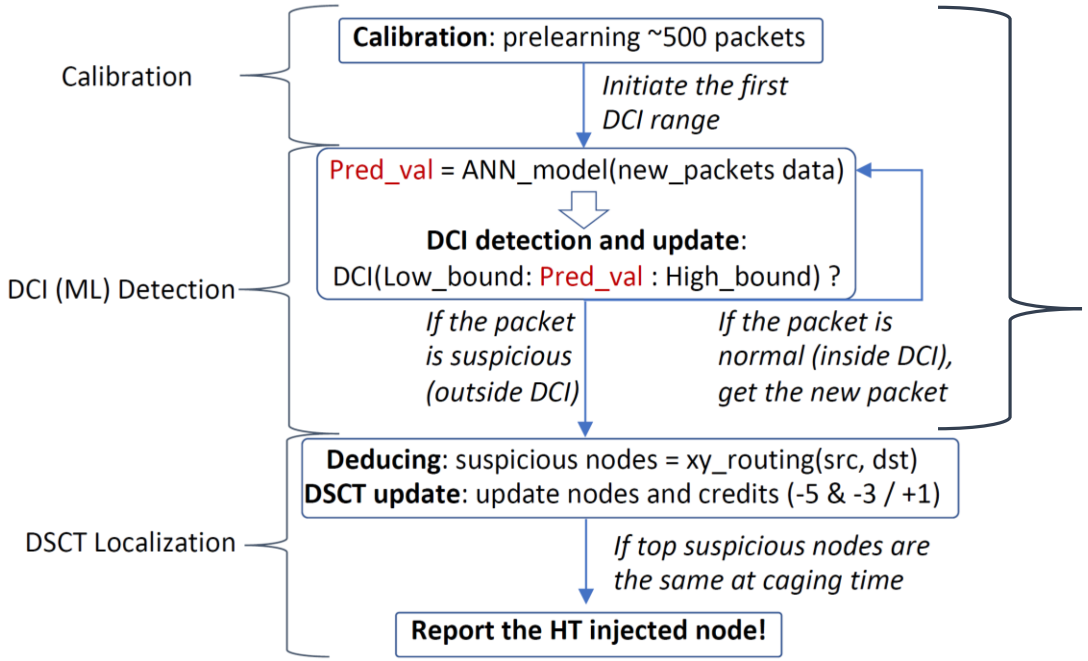


Fig4. Framework workflow

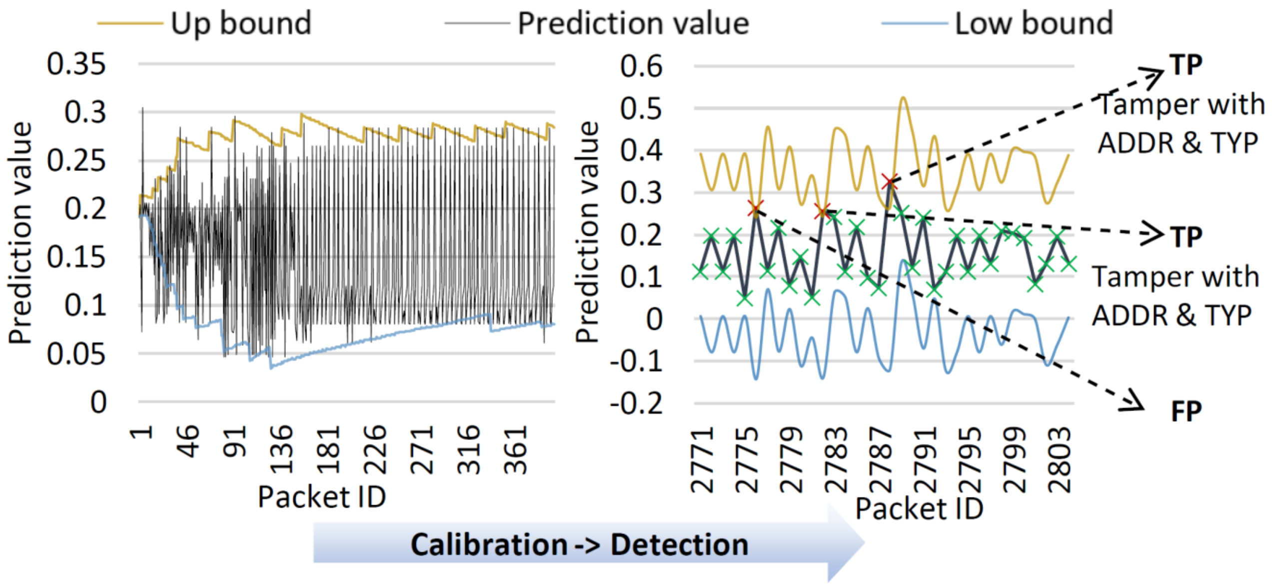


Fig5. DCI: Calibration and Detection

Proposed framework : DCI & ML model

- **Trained ML model**: ANN (2 hidden layers)
- **Features**: Address, Instruction Type, Source ID, Destination ID, Number of dependencies
- **Dataset**: Blackschole workload of PARSEC benchmark, parsed by Netrace tool

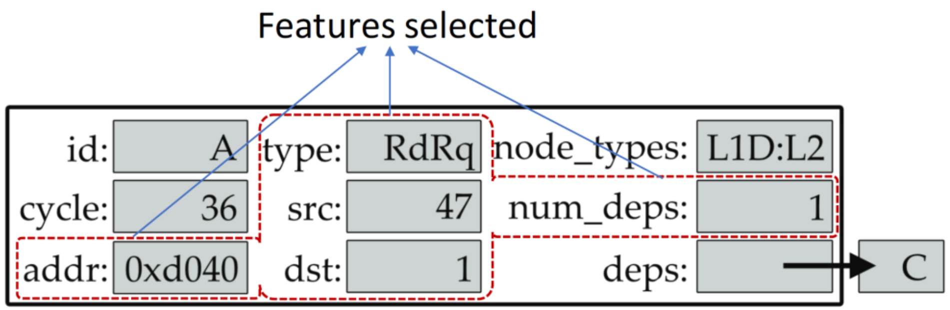


Fig6. Packet example

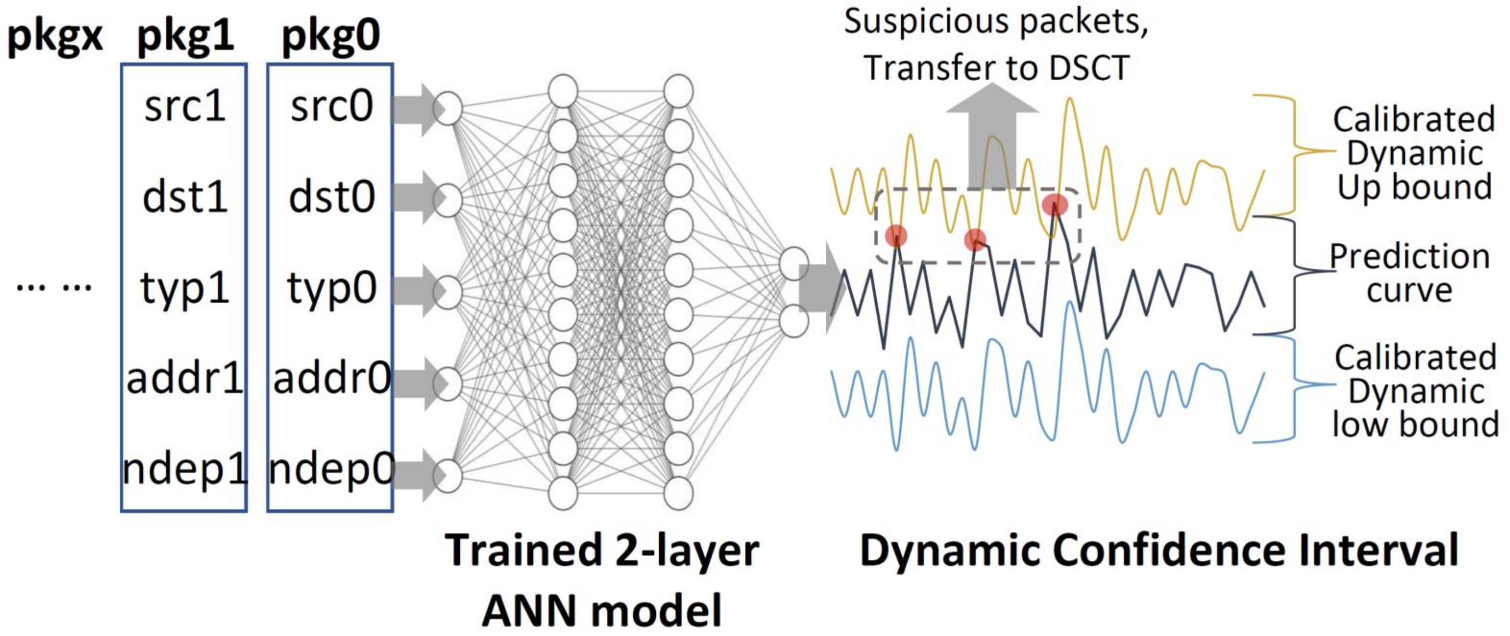


Fig7. DCI working with ANN

Proposed framework: DSCT localization illustration

- HT node localization: Dynamic Security Credit Table (DSCT)

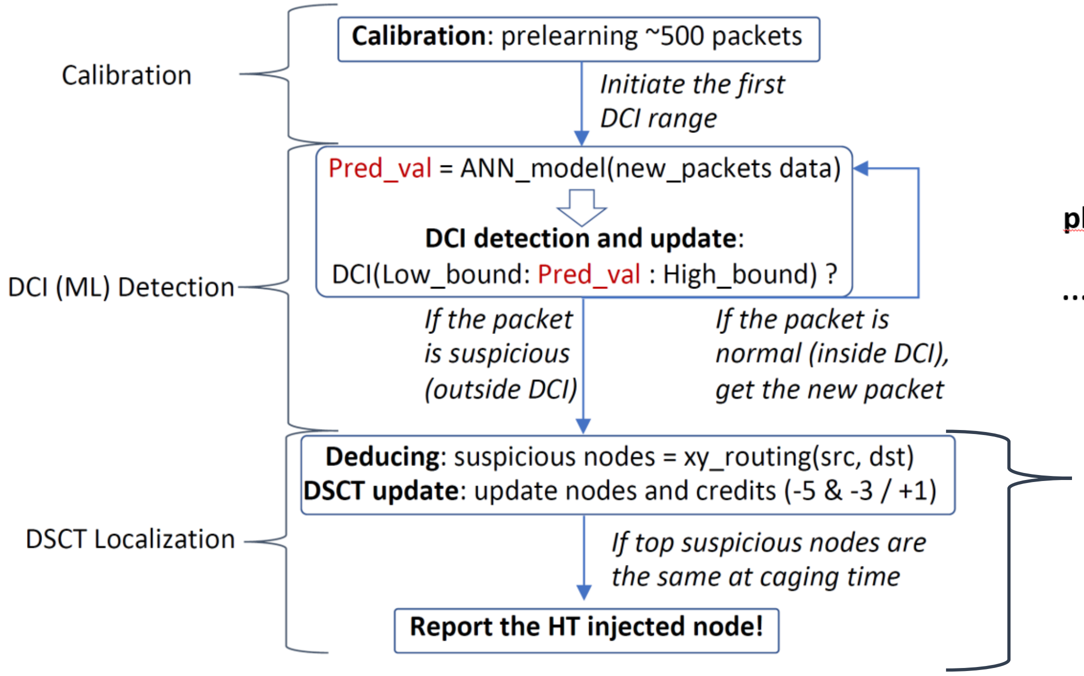


Fig8. Framework workflow

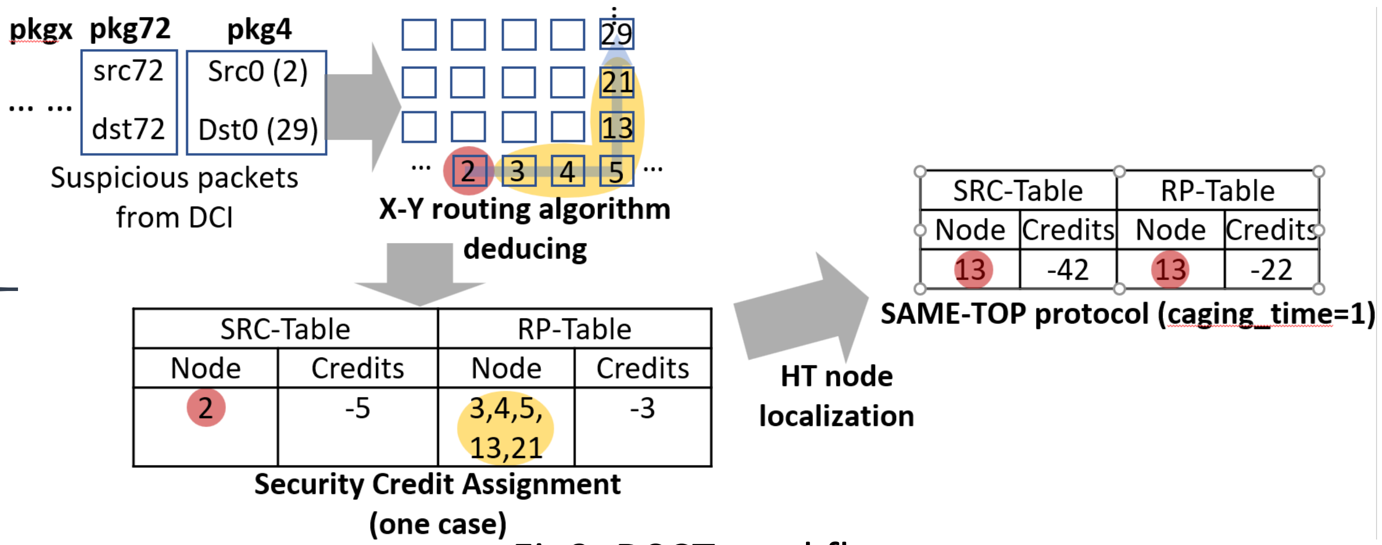


Fig9. DSCT workflow

Proposed framework: DSCT localization real example

- HT node localization: Dynamic Security Credit Table (DSCT)

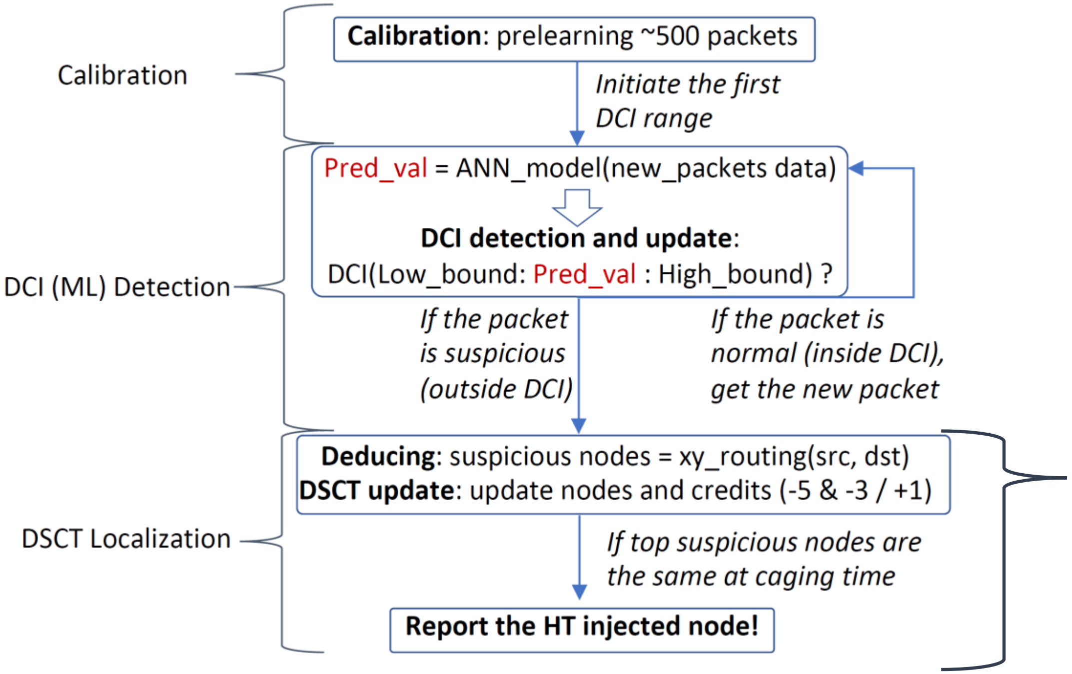
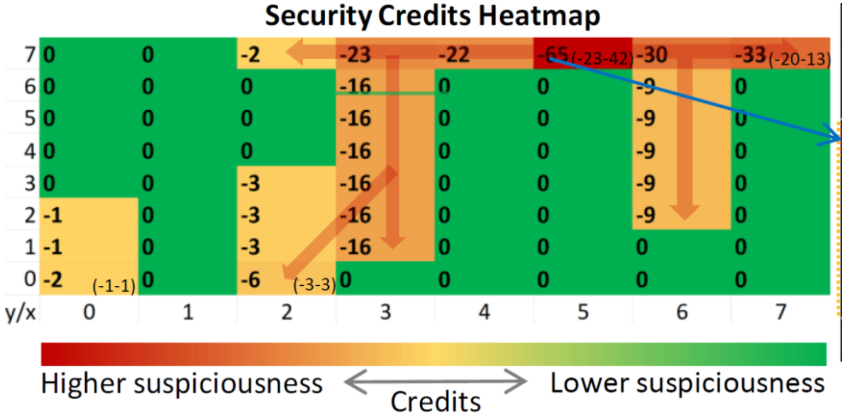


Fig10. Framework workflow



DSCT (source)	
Node (x,y)	Credit
61 (5,7)	-23
63 (7,7)	-20
2 (2,0)	-3
58 (2,7)	-2
0 (0,0)	-1
12	0

DSCT (routing-path)	
Node (x,y)	Credit
61 (5,7)	-42
62 (6,7)	-30
59 (3,7)	-23
60 (4,7)	-22
51 (3,6)	-16
43 (3,5)	-16
63 (7,7)	-13
…	…
32	0

When caging time = 5: check the worst-credit node in both DSCTs.

The same worst-credit node in both DSCTs has treated as HT injected.

Removed nodes with 0 credit (temporary no suspiciousness)

Fig11. DSCT Localization Example

Experiment and Results

- NoC configuration:

# of cores	64 O3CPUs
NoC Topology	8 x 8 2D Mesh
NoC Routing Algorithm	X-Y Routing
NoC Packet Length	168 Bits
NoC Packets Generator	Netrace
PARSEC Workload	Balckscholes (simsmall)

- Experiment flow:

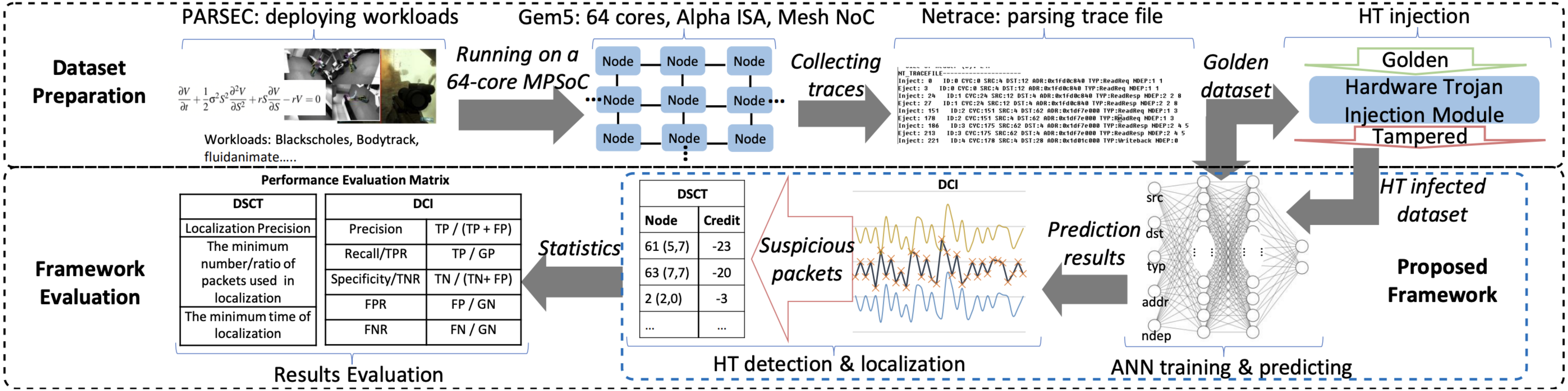


Fig12. Experiment flow

Experiment and Results

- DCI detection:

DSCT localization:

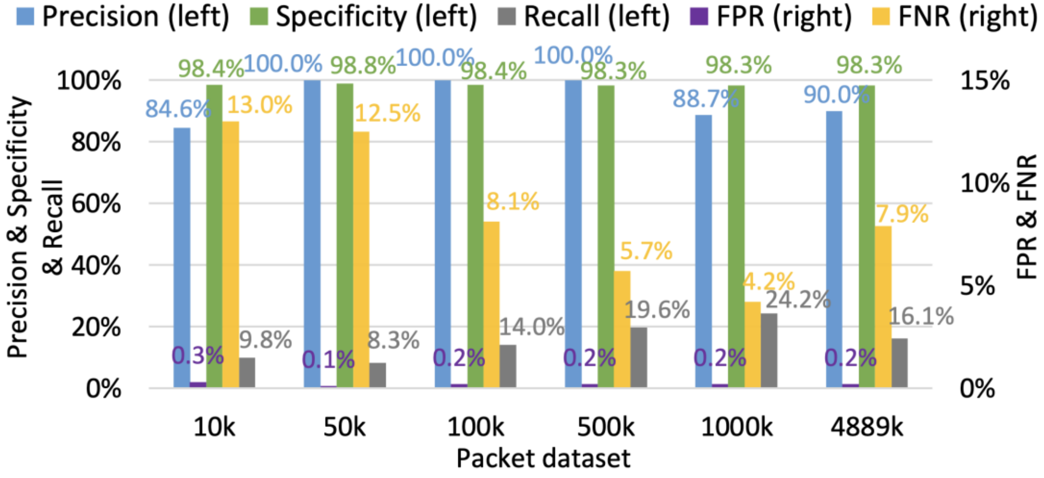


Fig13. Malicious packets (tampering attack) detection result

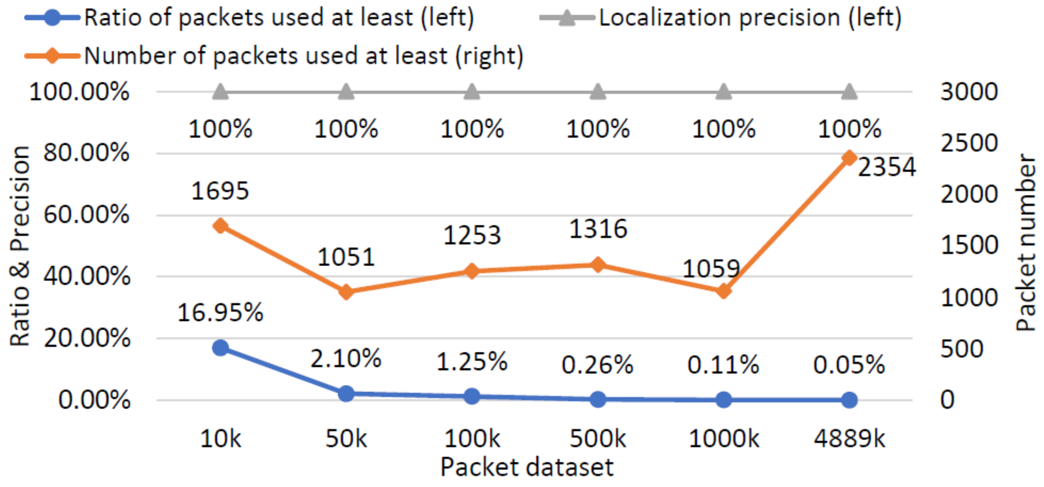


Fig14. HT-injected node localization result

- Comparison with related works:

	Charles et al. (2020)	Sinha et al. (2021b)	Chaves et al. (2019)	**[Our work]**
HT & Attacks	DoS: Flooding	DoS: Flooding	DoS: Path Collision	DoS: Packet Tampering
ML model	N/A	Perceptron -based ML	N/A	ANN
Detection Method	PAC, DLC	ML using BWT, IFI, VCL	CPRD Architecture	ML + DCI Algorithm
Precision	N/A	97.6 %	N/A	**96.3 %**
Localization Method	Event Handler for Router	MIP Algorithm	CPDD Architecture	DSCT Algorithm
Precision	N/A	96.7 %	N/A	**100 %**
Min-time	8∼24us @ 1.4GHz	30∼140 Cycles	97∼1118 Cycles	**5.8∼12.9us @ 2GHz**

Conclusion and Future Work

- First work to detect and localize tampering attack using ML
- Expected detection and localization precision and speed
- Future work1 for framework enhancement: A malicious node localization-specific workload/traffic pattern will be required instead of an application-specific workload (PARSEC).
- Future work2 for exploration: More SoC architectures could be explored, such as AMBA bus-based SoC system.

Reference

[1] Subodha Charles and Prabhat Mishra. A survey of network-on-chip security attacks and countermeasures. ACM Computing Surveys (CSUR), 54(5):1–36, 2021.

[2] Amey Kulkarni, Youngok Pino, Matthew French, and Tinoosh Mohsenin. Real-time anomaly detection framework for many-core router through machine-learning techniques. ACM Journal on Emerging Technologies in Computing Systems (JETC), 13 (1):1–22, 2016.

YOUR QUESTIONS