Anti-AdvTamp: Exposing Adversarial Evasive Tampering Attacks in Network-on-Chips with a Multi-Scale Attribute Fusion Detection

Shengkai Hu
School of ECS
University of Southampton
Southampton, UK
Shengkai.Hu@soton.ac.uk

Boojoong Kang
School of ECS
University of Southampton
Southampton, UK
b.kang@soton.ac.uk

Basel Halak
School of ECS
University of Southampton
Southampton, UK
basel.halak@soton.ac.uk

***Abstract*—Network-on-chip (NOC) architectures are increasingly becoming a widely adopted system-on-chip (SoC) architecture due to their scalable and modular communication structure, which matches the requirements of modern multiprocessor system-on-chip (MPSoC) and deep learning (DL) accelerator architectures, yet they are vulnerable to routing tampering attacks (e.g., traffic diversion) that induce Denial-of-Service (DoS). Therefore, machine learning (ML) have been increasingly applied to tampering detection in NoC; however, existing ML-based detectors struggle to cope with more covert adversarial tampering attacks. To address this challenge, this paper proposes SEAA, a symmetry-exploiting framework to construct two adversarial tampering attack models (AdvTamp1 and AdvTamp2), and evaluates their impact on ML-based detection. Furthermore, this research proposes a novel ML detection framework based on multi-scale attribute fusion, which integrates coarse-grained and fine-grained NoC performance metrics to improve detection accuracy. Experiments under PARSEC and synthetic Traffic Pattern benchmarks on various NoC topologies show that traditional ML models achieve detection accuracies of only 57.4% and 49.4% under AdvTamp1 and AdvTamp2 attacks, respectively. In contrast, our proposed multi-scale attribute fusion method significantly improves detection performance, achieving accuracies of 96.19% and 98% under AdvTamp1 and AdvTamp2 , respectively. These results demonstrate the effectiveness of the proposed method in improving detection under adversarial tampering and highlight its potential for enhancing the security of NoC.**

***Index Terms*—NoC, MPSoC, DL, Hardware Security, Hardware Trojan, DoS, Tampering, Adversarial Attack, Multi-Scale, Coarse-grained, Fine-grained, Machine Learning**

I. Introduction

In recent years, with the rapid advancement of machine learning and high-performance computing, Multiprocessor System-on-Chip (MPSoC) architectures have become increasingly prevalent. The Network-on-Chip (NoC) serves as a mainstream communication structure within MPSoCs, responsible for connecting the Central Processing Unit, memory, Intellectual Properties (IP), and other important modules, thereby facilitating data transmission and exchange within the MPSoC. NoCs have been widely adopted in Deep Learning (DL) accelerators, such as the one introduced by NVIDIA [30], and have been the subject of extensive performance optimization studies [29]. Beyond performance optimization, NoC has also been extensively studied in EDA and architecture communities as a system-level communication substrate, covering NoC-aware design methodologies (e.g., topology generation and layout-aware planning) [2] and validation/test infrastructures built upon NoC fabrics [31].

Given that the NoC serves as the access medium to all components and data [1], attackers have begun to target NoCs with Denial of Service (DoS) attacks. A DoS attack in NoC typically aims to "exploit shared resources to obstruct the transmission of legitimate traffic" [3]. One common approach is flooding, where attackers overwhelm the system with a high volume of requests or malicious packets [4], [6], [25], [26], leading to deadlocks or performance degradation. Another critical vector is routing tampering, such as Traffic Diversion and Traffic Loop attacks [5]. These tampering-based attacks achieve DoS by maliciously redirecting flits to non-optimal paths or black holes, resulting in packet loss and resource occupation without necessarily relying on high-intensity injection.

Detection of these tampering and DoS attacks typically relies on monitoring performance metrics such as packet flow and latency. Early methods [8]–[10] used threshold-based anomaly detection, but they struggle to adapt to dynamic environments. To address this, researchers have increasingly adopted machine learning (ML) [5]–[7], [11], [12]. However, existing ML-based approaches primarily focus on traditional, high-intensity attack patterns and fail to address covert adversarial tampering variants. This motivates exploring adversarially crafted tampering behaviors that can evade such detectors. We define AdvTamp as a stealthy variant of routing tampering that uses adversarial strategies to craft traffic patterns mimicking normal behavior, making it difficult for ML-based methods to detect [5]–[12].

To address these gaps, we propose SEAA to construct adversarial tampering (AdvTamp) attacks in NoC systems and develop a multi scale attribute fusion method to improve ML-based detection robustness. Our key contributions as follows:

- We propose SEAA, a framework that constructs Adv-

Tamp attacks by exploiting NoC topological symmetry. We developed AdvTamp1/2 variants that evade detectors trained on coarse datasets (e.g., [11]), limiting their detection accuracy to only 57.4% and 49.4%, respectively. SEAA is fully integrated into Gem5 with flexible command-line activation.
- We introduced a multi-scale attribute fusion method combining coarse-grained and fine-grained NoC attributes. The coarse-grained attributes targets conventional tampering attacks, while the fine-grained one detects evasive tampering attacks. Our method improves detection accuracy to 94% for AdvTamp1 and 97% for AdvTamp2.
- In the experiment, we simulated NoCs with 4 different topologies and used synthetic Traffic Pattern and PARSEC benchmarks to evaluate NoC performance metrics, thereby validating the effectiveness of our work.

II. Background

Network-on-Chip (NoC) is widely adopted as the on-chip communication fabric in modern MPSoCs and DL accelerators, enabling scalable interconnection among heterogeneous IP blocks. From the runtime viewpoint, NoC operation is shaped by multiple interacting factors such as topology and routing strategies, flow control and buffering, arbitration policies, and workload traffic characteristics. These factors are commonly reflected in observable indicators (e.g., packet injection/receipt behavior, latency-related statistics, buffer occupancy, and link utilization), which provide practical signals for runtime monitoring and identification.

As NoC increasingly becomes central to system integration, its correct and reliable operation also raises broader trust and security concerns. Recent research has investigated complementary directions to address security risks in NoC fabrics. SeVNoC provides a scalable RTL level validation scheme for systematically detecting security violations in inter-IP communications over NoC fabrics [15]. From the threat perspective, CONCEAL demonstrates covert NoC exploitation via stealthy flooding in IMC-based DNN accelerators, highlighting that NoC attacks can remain difficult to detect while causing substantial system-level degradation [16]. From the protection perspective, ObNoCs proposes an infrastructure to protect NoC fabrics against reverse-engineering attacks by enabling post fabrication programmability of inter-router connectivity [17].

Among the diverse security risks in NoCs [13], [14], Denial-of-Service (DoS) is particularly prevalent as it disrupts legitimate packet delivery by abusing shared communication resources. While traditional DoS is often associated with high-intensity traffic flooding designed to overwhelm network capacity, malicious routing tampering, such as traffic diversion, presents a more subtle threat. By manipulating flit movements to induce non-optimal paths or resource contention, tampering achieves a denial-of-service impact. Traditional non-machine-learning methods, such as functional testing and side-channel analysis [18], have been explored for detecting such anomalies; however, their effectiveness often degrades in complex NoC environments where these subtle malicious signals can be easily masked by inherent system noise.

With the advent of machine learning, significant progress has been made in NoC security monitoring. The study in [5] simulated representative threats, including routing tampering (e.g., Traffic Diversion) and flooding, demonstrating that selecting informative NoC metrics can enable effective ML-based detection. Furthermore, specialized frameworks have been developed to combat hardware-level threats; for instance, [32] proposed a distributed monitoring architecture using ML to detect data tampering and localize Hardware Trojans with near-perfect precision. Similarly, [12] utilized ANN-based detectors for attack localization, further highlighting the critical role of NoC indicator selection in identifying malicious behaviors.

Detecting stealthy adversarial DoS attacks remains challenging. Adversarial examples show that small perturbations can mislead ML models [19]; adversarially crafted tampering behaviors (AdvTamp) in NoCs can deliberately mimic benign runtime patterns to evade conventional detectors. This inherent vulnerability motivates our study on how adversarial evasion strategies degrade the reliability of state-of-the-art ML-based monitors. Furthermore, we explore how a multi-scale attribute fusion approach can expose these sophisticated threats, leading to the development of our robust Anti-AdvTamp detection framework.

III. Threat Model

We adopt a simplified threat model commonly used in NoC security studies, focusing on risks posed by malicious third-party IP (M3PIP) cores [3], [27]. We assume the adversary can compromise at least one untrusted on-chip IP during supply-chain integration (e.g., by implanting a hardware Trojan) and activate tampering behaviors under predefined trigger conditions (e.g., a specific transfer direction or a counter threshold) [5], [11], [32]. Concretely, the attacker can selectively drop packets or divert them to unintended destinations (i.e., nodes other than a packet's original intended destination), undermining the availability and functional correctness of on-chip communication and breaking end-to-end delivery semantics. This is particularly damaging in deep-learning accelerators, where missing or misdelivered weight/feature-map packets can stall computation or corrupt results [28], ultimately leading to DoS.

IV. SEAA: Symmetry-Exploiting Adversarial Attack Design

Traditional adversarial attacks operate at the software level by perturbing input data, whereas NoC-based tampering attack detection relies on run-time architectural metrics such as latency, link utilization, and hop count, which cannot be directly manipulated. To bridge this gap, we introduce **SEAA**, a symmetry-exploiting framework for constructing evasive adversarial tampering (AdvTamp) attacks in NoC systems. Using this framework, we design **AdvTamp1** and **AdvTamp2**, and instantiate them on representative regular NoC topologies

Algorithm 1: Time-Triggered SEAA Tampering Attacks (AdvTamp1/AdvTamp2)

Input: topology τ, mesh size (W, H), nodes N, source A, victim B, period T, ON length L, global window $[t_s, t_e]$, $mode \in \{\texttt{AdvTamp1}, \texttt{AdvTamp2}\}$

```
// Step 1: Compute symmetry-preserving
   remap target C
if τ is mesh then
    (x_a, y_a) ← (A mod W, ⌊A/W⌋);;
    (x_b, y_b) ← (B mod W, ⌊B/W⌋);;
    (x_c, y_c) ← (2x_a − x_b, 2y_a − y_b);;
     // Ensure C is within topology bounds;
    C ← x_c + y_c · W;;
else
    C ← (2A − B) mod N;;
    if C ∈ {A, B} then
        C ← (C + 1) mod N
    end
    ;
end
// Step 2: Stealthy destination remapping
   at source A
foreach packet p generated at cycle t by source s do
    if s ≠ A or t ∉ [t_s, t_e] or ¬in_on_window(t, T, L)
     then
        forward(p);
        continue;
    end
    d ← dst(p);
    // Unified remapping rule:
    // AdvTamp1: B → C
    // AdvTamp2: B ↔ C
    if d = B then
        modify_dst(p, C);
    end
    else if mode = AdvTamp2 and d = C then
        modify_dst(p, B);
    end
    forward(p);
end
```

in this work, namely mesh (Mesh_XY/Mesh_westfirst) and crossbar (Crossbar/CrossbarGarnet). Both attacks leverage the inherent symmetry of regular topologies to introduce subtle, symmetry-preserving path perturbations that keep key performance metrics statistically close to normal traffic, allowing them to statistically indistinguishable to ML-based detectors while still degrading NoC performance.

A. *AdvTamp1 (Symmetric Redirection):*

Mesh topologies exhibit strong geometric symmetry, where multiple destinations can have identical Manhattan (hop) distances from a given source under deterministic routing, while crossbar topologies provide cost-equivalent paths from a source to any destination. Leveraging this regularity, we instantiate AdvTamp1 on Mesh-XY as an illustrative example via symmetry-preserving destination remapping. As specified in Algorithm 1, during the randomly selected ON windows, the compromised source at node A redirects the destination of packets originally targeting B to its symmetric counterpart $C = \mathrm{Sym}(A, B)$, i.e., $B \to C$. As illustrated in Fig. 1(a), an example mapping is $A = N3$, $B = N14$, and $C = N4$.

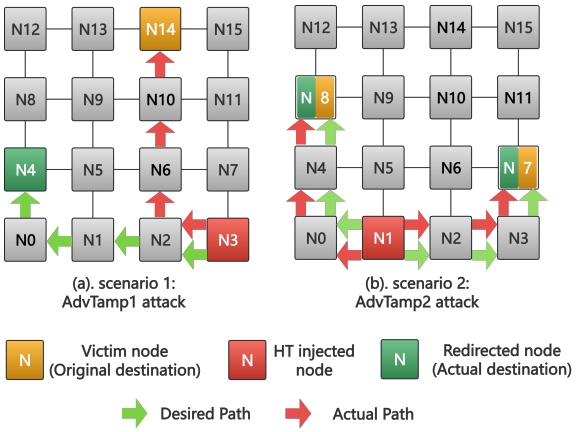

Fig. 1 AdvTamp attacks scenarios

Since B and C have equal hop distance from A under XY routing, the remapped packets traverse statistically similar paths, resulting in only subtle changes in aggregated KPIs while degrading intended delivery.

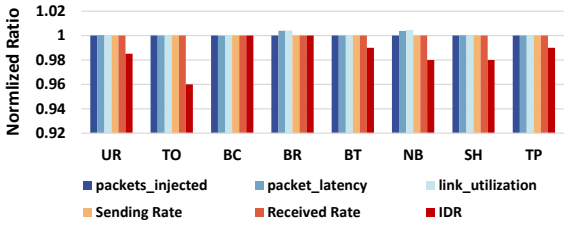

Fig. 2 NoC performance veriation under AdvTamp1 attack

As shown in Fig. 2, AdvTamp1 induces only subtle changes in the NoC key performance indicators (KPIs): the normalized injection/sending rate, mean packet latency, average hop count, and link utilization remain close to the baseline across all eight traffic patterns (Uniform Random (UR), Tornado (TO), Bit Complement (BC), Bit Reverse (BR), Bit Rotation (BT), Neighbor (NB), Shuffle (SH), and Transpose (TP)). However, the *Intended Delivery Ratio (IDR)* drops noticeably. Here, IDR is defined as the fraction of injected packets that are ultimately received at their *original intended destinations*. The reduced IDR indicates that a considerable portion of packets fails to reach their intended endpoints (i.e., misdelivery and/or effective loss), thereby violating end-to-end delivery semantics and causing DoS impact. Since global average KPIs exhibit limited deviation, the attack remains stealthy and is therefore harder for ML-based detectors to identify.

B. *AdvTamp2 (Address Swap):*

Following SEAA, AdvTamp2 instantiates a stronger symmetry-based perturbation via *destination swapping* between a symmetric node pair. As specified in Algorithm 1, during the randomly selected ON windows, packets injected from the compromised source at node A with destination B are remapped to $C = \mathrm{Sym}(A, B)$, while packets originally destined for C are remapped to B (i.e., $B \leftrightarrow C$). Fig. 1(b) illustrates an example swap between $B = N7$ and $C = N8$ under attacker $A = N1$. Since B and C are geometrically

symmetric with respect to A, the swapped traffic induces only subtle changes in KPIs while amplifying the degradation in intended delivery compared with AdvTamp1.

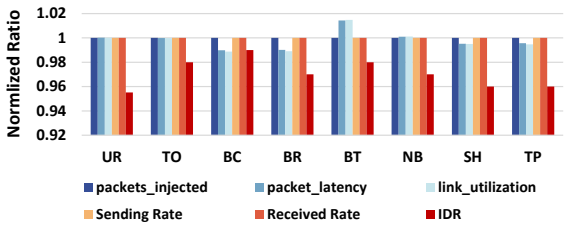

Fig. 3 NoC performance veriation under AdvTamp2 attack

Compared with Fig. 2, Fig. 3 shows that AdvTamp2 introduces more noticeable changes in both latency and link utilization across traffic patterns, while still keeping the normalized global KPIs within a relatively narrow range (0.94–1.02). More importantly, the *Intended Delivery Ratio (IDR)* exhibits a larger degradation, indicating that a higher fraction of packets is diverted away from their original intended destinations (i.e., misdelivery and/or effective loss). Consequently, AdvTamp2 causes a stronger DoS, yet remains challenging for ML-based detectors to identify due to the limited deviation in KPIs.

V. Enhanced Detection Model with Multi-Scale Attributes Fusion

To address the growing challenges posed by adversarial tampering (AdvTamp) attacks in NoC, we proposes a novel detection method based on multi-scale machine learning. Using the Gem5 simulator [20] integrated with the HeteroGarnet framework [21], we construct two 8×8 Mesh NoCs (`Mesh_XY` and `Mesh_westfirst`) and two 64-core Crossbar NoCs (`Crossbar` and `CrossbarGarnet`), covering diverse topologies for comprehensive evaluation. We then simulate AdvTamp1 and AdvTamp2 attacks under the four PARSEC benchmarks and all Traffic Pattern benchmarks.

We build a multi-scale attribute fusion dataset spanning temporal and spatial scales: coarse-grained metrics capture broad network performance fluctuations under conventional attacks, while fine-grained metrics expose subtle anomalies to boost detection precision. Using this dataset, we evaluate seven complementary machine learning models (SVM, KNN, RFC, DCT, NBC, LR, and MLP) to validate the generalizability and robustness of our approach across different models and diverse AdvTamp attack scenarios.

A. Multi-Scale Attributes Fusion Dataset

As shown in Fig. 4, we construct a multi-scale attributes fusion dataset that combines coarse-grained and fine-grained NoC observations to improve detection under diverse attack scenarios. Coarse-grained attributes provide a lightweight view of overall network performance and are widely adopted for runtime monitoring due to their low computation and hardware overhead [5], [7], [14], [18], [27]. However, AdvTamp variants may introduce only subtle shifts in statistics, making coarse-grained data alone insufficient for reliable detection. To address this limitation, fine-grained attributes offer complementary, localized evidence of traffic irregularities, enabling more precise and robust identification. By integrating both scales, the proposed dataset captures both global performance degradation and detailed behavioral signatures.

- The coarse-grained dataset includes Packets Injected, Packets Received, Average Packet Queueing Latency, Average Packet Network Latency, Average Packet Latency, Average Link Utilization, and Average Hops. These attributes effectively reflect the overall network performance and traffic conditions, making them suitable for detecting conventional tampering attacks [5], [11], [32].
- The fine-grained dataset includes Sending Rate, and Received Rate and Packet Loss Rate. These attributes allow for a more in-depth analysis of network behavior, ensuring accuracy and reliability in detection. Unlike coarse-grained dataset, these statistics describe delivery outcomes within a window (offered load, effective delivery), exposing localized and short-lived irregularities that may be smoothed out in coarse aggregation, and thus strengthening detection under stealthy variants.

Coarse-grained and fine-grained attributes provide complementary views across both temporal and observation granularity. Temporally, coarse-grained attributes summarize global NoC behavior at a lower sampling frequency, capturing long-term trends in performance and congestion, whereas fine-grained attributes are derived from packet event records (with flexible temporal resolution). From an observation perspective, coarse-grained metrics reflect globally aggregated network status, while fine-grained features are event derived traffic outcomes (e.g., sending and loss statistics). By combining both scales, we construct a multi-scale attributes fusion dataset that jointly models global degradation and delivery anomalies.

B. Benchmarking Multi-Scale Attributes Fusion

We selected seven machine learning models for AdvTamp attack detection: Support Vector Machine (SVM), K Nearest Neighbors (KNN), Random Forest (RFC), Decision Tree (DCT), Naive Bayes (NBC), Logistic Regression (LR), and Multi Layer Perceptron (MLP). These models were chosen based on their complementary strengths in processing multi-scale datasets. SVM and RFC are effective for medium sized data with high dimensional or nonlinear structures [22], while KNN and LR perform well on smaller or linearly separable datasets [23]. MLP can capture complex nonlinear patterns and is particularly suitable for fine grained analysis [24]. This diverse model selection enables us to validate the effectiveness of our multi-scale fusion method across a variety of ML.

VI. Experiment and Evaluation

We first normalized both coarse-grained and fine-grained attributes to eliminate scale differences across feature dimensions and facilitate model training. We then performed correlation analysis on the coarse-grained dataset (see Fig. 5). The results indicate a strong correlation between *Packets Injected* and the *Packet Latency*, suggesting that they capture closely related global performance variations under traffic perturbations.

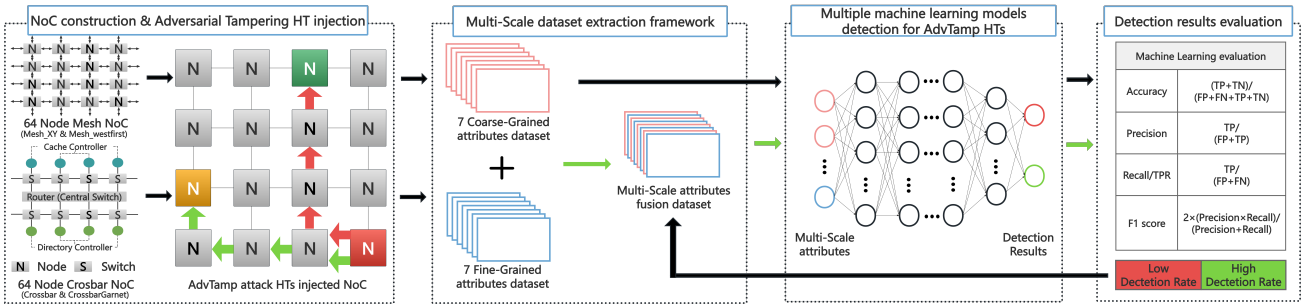

Fig. 4 Overview of the adopted experiment framework

As a key NoC metric, *Link Utilization* also exhibits significant correlations with other coarse-grained indicators, reflecting its sensitivity to congestion and performance degradation.

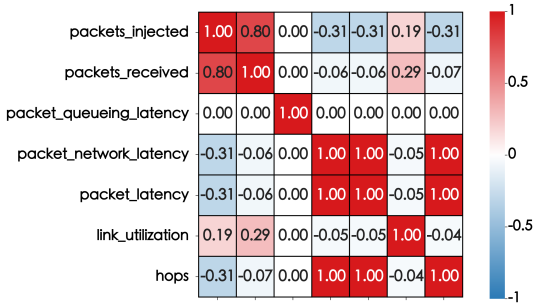

Fig. 5 Correlation Heatmap

Based on this analysis, we select three coarse-grained key performance attributes, namely *Packets Injected*, *Packet Latency*, and *Link Utilization*. Since *Packets Injected* already characterizes the offered load at the coarse level, we avoid introducing redundant fine-grained load indicators and instead choose two traffic-outcome features computed from packet-event statistics: *Packets Received Rate* and *Packet Loss Rate*. When constructing the fused dataset, we use the coarse-grained sampling period as a unified monitoring window, aggregate fine-grained packet events within each window, and align them temporally with the corresponding coarse-grained samples. The monitoring window length is set to T=100 cycles; therefore, a single simulation run for each topology–benchmark pair (lasting 1,000,000 cycles) yields 10,000 aligned samples. Attack windows are generated using a periodic duty-cycle scheme: within every 10,000-cycle interval, we randomly select an attack start time and keep the attack active for 500 consecutive cycles, resulting in an overall attack ratio of 5%.

A. Dataset Construction and Split Strategy

Our evaluation covers 4 NoC topologies (Mesh_XY, Mesh_westfirst, Crossbar, and CrossbarGarnet) and 12 benchmark configurations, including 8 synthetic traffic patterns (*bit complement*, *bit reverse*, *bit rotation*, *neighbor*, *shuffle*, *tornado*, *transpose*, and *uniform random*) and 4 PARSEC workloads (*blackscholes*, *bodytrack*, *canneal*, and *dedup*). For each topology–benchmark pair, we construct a window-level AdvTamp dataset with 10,000 samples using the same attack activation scheme described earlier (i.e., a fixed 5% attack-time ratio at the window level), and apply a 70%/30% train/test split within each dataset. This yields $4 \times 12 \times 10{,}000 = 480{,}000$ samples per case; considering four cases (coarse–AdvTamp1, coarse–AdvTamp2, multi–AdvTamp1, and multi–AdvTamp2), we evaluate a total of **1,920,000** samples.

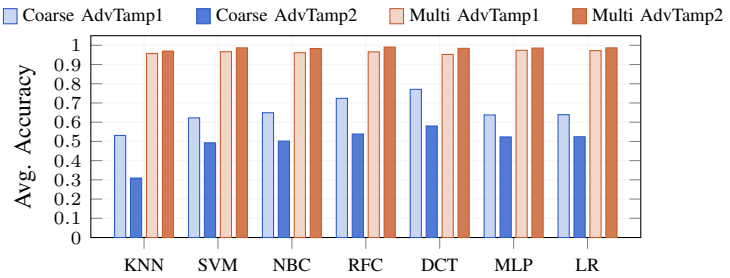

Fig. 6 Average detection accuracy across all workloads.

B. Evaluation on Coarse-Grained Dataset

Notably, the attributes used are consistent with [10], which demonstrated that coarse-grained datasets are highly effective for detecting conventional NoC tampering and DoS attacks. While those methods achieve over 95% average accuracy for threats like traffic diversion, traffic loops, and flooding, our results in Figure 6 show a performance collapse against adversarial variants. The average detection accuracy for AdvTamp1 and AdvTamp2 drops to 57.4% and 49.4%, respectively. This stark contrast proves that while traditional coarse-grained monitoring is robust against conventional threats [10], it is successfully bypassed by AdvTamp, necessitating our proposed multi-scale fusion framework.

C. Evaluation on Multi-Scale Attribute Fusion

Using the multi-scale attribute fusion dataset, we observed a significant improvement in detection accuracy. As shown in Figure 6, all models achieved over 95% accuracy, with AdvTamp2 averaging 96.14%. In traffic pattern scenarios like transpose and shuffle, MLP even achieved 100% detection accuracy. Similarly, for PARSEC, models showed exceptional performance in detecting AdvTamp2, with detection accuracy exceeding 97% for all workloads. These results confirm that multi-scale attribute fusion significantly enhances detection performance and improves the detection of AdvTamp attacks.

We also evaluated multi-scale attribute fusion across models by comparing precision, recall, and F1 score (Table I). Both AdvTamp1 and AdvTamp2 achieved precision above 0.98, with recall at 0.91 and 0.96 respectively. AdvTamp2's high

TABLE I Multi-Scale dataset detection average metrics. Values > 0.95 are highlighted in blue.

	Precision		Recall		F1 Score	
Model	Adv1	Adv2	Adv1	Adv2	Adv1	Adv2
KNN	0.99	0.99	0.89	0.92	0.93	0.96
SVM	0.99	1.00	0.91	0.97	0.95	0.98
NBC	1.00	1.00	0.89	0.95	0.94	0.97
RFC	0.96	0.99	0.94	0.98	0.95	0.99
DCT	0.93	0.97	0.94	0.98	0.93	0.98
MLP	1.00	1.00	0.93	0.98	0.96	0.99
LR	1.00	1.00	0.92	0.96	0.96	0.98
Average	0.98	0.99	0.91	0.96	0.94	0.98

F1 score of 0.98, together with its very low false positive and false negative rates, demonstrates the robust capability of our method in neutralizing both variants of AdvTamp attacks.

VII. Conclusion

This paper studies adversarial tampering (AdvTamp) threats in Network-on-Chip (NoC) systems and proposes a multi-scale attribute fusion detection framework. Experiments on various NoC topologies under PARSEC and synthetic traffic benchmarks show that, under AdvTamp1 and AdvTamp2, classic ML-based detectors achieve only 57.4% and 49.4% detection accuracy, respectively, when using coarse-grained indicators. By integrating coarse-grained and fine-grained NoC performance metrics, the proposed multi-scale fusion method achieves 96.19% accuracy for AdvTamp1 and 98% for AdvTamp2, with precision, recall, and F1-score all exceeding 90%. Future work will extend this study to other adversarial attacks and conduct hardware-level performance evaluations.

References

[1] R. Js, D. M. Ancajas, K. Chakraborty, and S. Roy, "Runtime Detection of a Bandwidth Denial Attack from a Rogue Network-on-Chip," in *Proc. 9th Int. Symp. on Networks-on-Chip*, Vancouver, BC, Canada, Sep. 2015, pp. 1–8. doi: 10.1145/2786572.2786580.

[2] B. Yu, S. Dong, S. Chen, *et al.*, "Floorplanning and Topology Generation for Application-Specific Network-on-Chip," in *2010 15th Asia and South Pacific Design Automation Conf. (ASP-DAC)*, Jan. 2010, pp. 535–540. doi: 10.1109/ASPDAC.2010.5419825.

[3] S. Charles, Y. Lyu, and P. Mishra, "Real-time Detection and Localization of DoS Attacks in NoC based SoCs," in *2019 Design, Automation & Test in Europe Conf. & Exhibition (DATE)*, Mar. 2019, pp. 1160–1165. doi: 10.23919/DATE.2019.8715009.

[4] C. G. Chaves, S. P. Azad, T. Hollstein, and J. Sepulveda, "A Distributed DoS Detection Scheme for NoC-based MPSoCs," in *2018 IEEE Nordic Circuits and Systems Conf. (NORCAS): NORCHIP and Int. Symp. System-on-Chip (SoC)*, Oct. 2018, pp. 1–6. doi: 10.1109/NORCHIP.2018.8573524.

[5] A. Kulkarni, Y. Pino, M. French, and T. Mohsenin, "Real-Time Anomaly Detection Framework for Many-Core Router through Machine-Learning Techniques," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 13, no. 1, pp. 1–22, Jan. 2017. doi: 10.1145/2827699.

[6] H. Wang, B. Halak, J. Ren, and A. Atamli, "DL2Fence: Integrating Deep Learning and Frame Fusion for Enhanced Detection and Localization of Refined Denial-of-Service in Large-Scale NoCs," *arXiv preprint arXiv:2403.13563*, 2024.

[7] K. Wang, H. Zheng, and A. Louri, "TSA-NoC: Learning-Based Threat Detection and Mitigation for Secure Network-on-Chip Architecture," *IEEE Micro*, vol. 40, no. 5, pp. 56–63, Sep. 2020. doi: 10.1109/MM.2020.3003576.

[8] H. Khattri, N. K. V. Mangipudi, and S. Mandujano, "HSDL: A Security Development Lifecycle for Hardware Technologies," in *2012 IEEE Int. Symp. Hardware-Oriented Security and Trust (HOST)*, Jun. 2012, pp. 116–121. doi: 10.1109/HST.2012.6224330.

[9] H. Salmani, "COTD: Reference-Free Hardware Trojan Detection and Recovery Based on Controllability and Observability in Gate-Level Netlist," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 2, pp. 338–350, Feb. 2017. doi: 10.1109/TIFS.2016.2613842.

[10] K. Xiao and M. Tehranipoor, "BISA: Built-in Self-Authentication for Preventing Hardware Trojan Insertion," in *2013 IEEE Int. Symp. Hardware-Oriented Security and Trust (HOST)*, Jun. 2013, pp. 45–50. doi: 10.1109/HST.2013.6581564.

[11] S. Hu, H. Wang, and B. Halak, "Cascaded Machine Learning Model Based DoS Attacks Detection and Classification in NoC," in *2023 IEEE Int. Symp. Circuits and Systems (ISCAS)*, Monterey, CA, USA, May 2023, pp. 1–5. doi: 10.1109/ISCAS46773.2023.10182218.

[12] H. Wang and B. Halak, "Hardware Trojan Detection and High-Precision Localization in NoC-Based MPSoC Using Machine Learning," in *Proc. 28th Asia and South Pacific Design Automation Conf. (ASPDAC '23)*, New York, NY, USA, Jan. 2023, pp. 516–521. doi: 10.1145/3566097.3567922.

[13] A. Sarihi *et al.*, "A Survey on the Security of Wired, Wireless, and 3D Network-on-Chips," *IEEE Access*, vol. 9, pp. 107625–107656, 2021. doi: 10.1109/ACCESS.2021.3100540.

[14] T. Boraten and A. K. Kodi, "Mitigation of Denial of Service Attack with Hardware Trojans in NoC Architectures," in *2016 IEEE Int. Parallel and Distributed Processing Symposium (IPDPS)*, May 2016, pp. 1091–1100. doi: 10.1109/IPDPS.2016.59.

[15] X. Meng, K. Raj, S. Ray, *et al.*, "SeVNoC: Security Validation of System-on-Chip Designs with NoC Fabrics," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 42, no. 2, pp. 672–682, Feb. 2023. doi: 10.1109/TCAD.2022.3179307.

[16] S. Das, S. Kundu, S. K. Mandal, *et al.*, "CONCEAL: Covert NoC Exploitation in In-Memory Computing-based DNN Accelerators," in *2025 IEEE Int. Conf. on Omni-layer Intelligent Systems (COINS)*, 2025, pp. 1–7. doi: 10.1109/COINS65080.2025.11125783.

[17] D. Halder, M. Merugu, and S. Ray, "ObNoCs: Protecting Network-on-Chip Fabrics Against Reverse-Engineering Attacks," *ACM Trans. Embedded Comput. Syst.*, vol. 22, no. 5s, pp. 1–21, Sep. 2023. doi: 10.1145/3609107.

[18] A. Adamov, A. Saprykin, D. Melnik, and O. Lukashenko, "The Problem of Hardware Trojans Detection in System-on-Chip," in *2009 10th Int. Conf. on the Experience of Designing and Application of CAD Systems in Microelectronics*, Lviv, Ukraine, 2009, pp. 178–179.

[19] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," *arXiv*, Mar. 2015. doi: 10.48550/arXiv.1412.6572.

[20] N. Binkert *et al.*, "The gem5 Simulator," *ACM SIGARCH Comput. Arch. News*, vol. 39, no. 2, pp. 1–7, 2011.

[21] S. Bharadwaj *et al.*, "Kite: A Family of Heterogeneous Interposer Topologies Enabled via Accurate Interconnect Modeling," in *2020 57th ACM/IEEE Design Automation Conf. (DAC)*, IEEE, 2020, pp. 1–6.

[22] S. Shalev-Shwartz, Y. Singer, and N. Srebro, "Pegasos: Primal Estimated Sub-Gradient Solver for SVM," in *Proc. 24th Int. Conf. on Machine Learning*, 2007, pp. 807–814.

[23] S. Zhang, "Challenges in KNN Classification," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 10, pp. 4663–4675, 2021.

[24] A. Botalb *et al.*, "Contrasting Convolutional Neural Network (CNN) with Multi-Layer Perceptron (MLP) for Big Data Analysis," in *2018 Int. Conf. on Intelligent and Advanced System (ICIAS)*, IEEE, 2018, pp. 1–5.

[25] M. S. Hathal *et al.*, "Attack and Anomaly Prediction in Networks-on-Chip of MPIoT-Based SoC Utilizing Machine Learning Approaches," *Service Oriented Comput. Appl.*, vol. 18, no. 3, pp. 209–223, 2024.

[26] Z. Pan *et al.*, "Hardware-Assisted Malware Detection Using Machine Learning," in *2021 Design, Automation & Test in Europe Conf. & Exhibition (DATE)*, IEEE, 2021, pp. 1775–1780.

[27] X. Chen, Q. Liu, S. Yao, J. Wang, and Y. Zhao, "Hardware Trojan Detection in Third-Party Digital Intellectual Property Cores by Multi-level Feature Analysis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 7, pp. 1370–1383, 2017.

[28] Y. Chen, T. Luo, S. Liu, *et al.*, "DaDianNao: A Machine-Learning Supercomputer," in *Proceedings of the 47th Annual IEEE/ACM International Symposium on Microarchitecture*, pp. 609–622, 2014.

[29] W. Zhu, Y. Chen, and Z. Lu, "NoCDAS: A Cycle-Accurate NoC-Based Deep Neural Network Accelerator Simulator," ACM Trans. Model. Comput. Simul., 2025.

[30] Y. S. Shao, J. Clemons, R. Venkatesan, et al., "Simba: Scaling deep-learning inference with multi-chip-module-based architecture," in *Proc. 52nd IEEE/ACM Int. Symp. Microarchitecture (MICRO)*, 2019, pp. 14–27.

[31] F. Yuan, L. Huang, and Q. Xu, "Re-Examining the Use of Network-on-Chip as Test Access Mechanism," in *Proc. of the Conf. on Design, Automation and Test in Europe (DATE)*, Mar. 2008, pp. 808–811. doi: 10.1109/DATE.2008.4484917.

[32] H. Wang and B. Halak, "TampML: Tampering Attack Detection and Malicious Nodes Localization in NoC-Based MPSoC," *IEEE Trans. Emerging Topics Comput.*, vol. 13, no. 2, pp. 551–562, Apr.–Jun. 2024. doi: 10.1109/TETC.2024.3374465.