Cascaded Machine Learning Model Based DoS Attacks Detection and Classification in NoC

Shengkai Hu
School of Electronics and Computer Science
University of Southampton
Southampton, United Kindom
sh4u21@soton.ac.uk

Haoyu Wang
School of Electronics and Computer Science
University of Southampton
Southampton, United Kindom
haoyu.wang@soton.ac.uk

Basel Halak
School of Electronics and Computer Science
University of Southampton
Southampton, United Kindom
Basel.Halak@soton.ac.uk

***Abstract*—Network-on-Chip (NoC) is becoming an increasingly common System-on-Chip (SoC) fabric architecture since it matches the characteristics of the SoC's shared storage and high-frequency communication. However, due to the rising utilization of NoC, a large number of adversaries are trying to inject Hardware Trojan (HT) into NoC to obtain profits. An increasing variety of NoC HTs is emerging and implemented, resulting in current detection methods becoming invalid. This paper presents a cascaded machine learning model based Denial-of-Service (DoS) attack detection and classification approach. An Support Vector Machine (SVM) and a K-Nearest Neighbor (KNN) model were employed in the framework, which has also been validated on our runtime mixed dataset consisting of normal and attacked data extracted from four traffic pattern cases. The proposed framework achieved an expected detection accuracy: more than 85% on detection in average. And outstanding classification results on every attack: 97% on Flooding, and up to 100% on both Routing Loop and Traffic Diversion.**

Keywords—Hardware Trojan, Machine Learning, NoC, DoS Attacks, Hardware Security

I. INTRODUCTION

Due to the widespread application of NoCs, their security priority should be gradually increased. Since hardware is the most privileged entity, being able to manipulate it can give attackers considerable flexibility and opportunities to launch malicious security attacks. Additionally, many hardware-oriented attacks have evasion capabilities through defensive detection scanners [1]. A System-on-Chip (SoC) is a chip that implements most or even the complete functionality of an electronic system inside a single chip. Chip Multiprocessors (CMP) is a relatively prevalent architecture. The CMP architecture is characterized by using shared storage to exchange data. Therefore, every core can know the entire address space. The structure of CMP shared storage facilitates data transmission between cores. But the possibility of HT insertion becomes higher in CMP. Such as routers, can easily be inserted by hardware Trojans, so the hardware security issues on NoC are gradually being studied. Our research focused on HT-initiated DoS attack detection in NoC-based CMP.

The Hardware Trojans (HT) simulated in this paper are activated under specific conditions. Condition-based Trojan activation is implemented in several router-to-router transmission internal logic states. Attacks on multi-core routers can impact network packet transfer rates, network/processing core availability, and disruption of core communications [2]. Routers can be attacked externally through memory fabric interfaces, dedicated core interfaces, or internally by corrupting routing tables. Includes Traffic Diversion, Routing Loop[3], and Flooding[4] attacks. All three HT-based attacks are also known as Denial of Service (DoS) attacks, in which certain routers are attacked and rendered unavailable.

The existing methods to enhance hardware security are mainly implemented in the following three stages: design stage, test stage, and operation stage. The primary defense method in the design stage is HSDL (Hardware Security Development Lifecycle) [5]. This method was mainly to design the overall hardware according to a reasonable design process. The primary defense method in the test stage is FHL (fault history) [6], which mainly refers to the past attack history for specific aspects of detection; The primary method of operation stage is RTM (Runtime threshold monitoring) [7]. This method is mainly to manually design some characteristic indicators of the hardware as thresholds and detect whether the set thresholds are exceeded when the hardware is running. To monitor whether the NoC is attacked, most of these traditional methods of detecting Trojans rely on historical records or manually adjusting thresholds and lack refined and intelligent detection of Trojan horses. As a result, it is necessary to use more thoughtful and advanced methods. In this paper, we innovatively proposed a method that takes advantage of a cascade of machine learning (ML) algorithms to detect and classify NoC attacks accurately. The main contributions of this paper are as follows:

- We proposed a scheme of DoS attacks detection and classification in NoC. It consists of an SVM ML model first to detect attacked feature variations and a KNN model to further classify the suspicious data from SVM output as different specific types of attacks or noise.
- We selected suitable NoC features using feature correlation analysis. In addition, various ML models were comprehensively explored to determine superior algorithms for our framework.
- We established an experiment setup using gem5 and Garnet [8] NoC simulator with the benchmark including different cases of traffic patterns. For less assumption, a mixed dataset

combining normal with three attacks on all traffic patterns has been developed and employed.

II. BACKGROUND AND PRELIMINARIES

A. HT-based attacks in NoCs

It has been proved that NoC's Hardware Trojan can infect NoC's link [9] and router micro-architecture [10]. Generally speaking, HT is implanted in the IC design layout [11]. The types of implanted HT can also be broadly divided into two categories: permanently activated and conditionally activated. As the name suggests, a permanently activated HT is always active and may contain transient faults at any time, affecting the normal transmission of the NoC. Condition-based Trojans are triggered under specific conditions, and after implantation, they generally remain dormant, which can fool general hardware detection until an attacker activates them. Either permanently activated HTs or conditionally activated HTs, when they are activated, can cause transient faults by forcing bit flips in the link or changing the transmission direction of the router. These failures can lead to massive retransmission traffic, address spoofing, and network saturation, which will render the NoC unusable.

B. Conventional techniques for detecting NoC HTs

It can be noted that the operation of state-of-the-art countermeasures against HT attacks in NoC can be divided into three phases: design phase, testing phase, and operation phase.

There are corresponding detection methods at each stage, among which the primary method used in the design stage is HSDL (Hardware Security Development Lifecycle) [5]. HSDL focuses on providing technology with security, and it is divided into five stages. At each stage, the designer manually designs the required input and output and then checks whether the NoC indicators during the design match the input and output manually designed by the designer to achieve the purpose of detecting HT. The method commonly used in the testing phase is FHL [6], which is a method to monitor the NoC attributes related to failures (such as temperature and buffer/link utilization) while the NoC is running. If any attribute values exceed their corresponding manually set thresholds, the system will mark the corresponding NoC component as HT-infected. Most of the detection methods at runtime are RTM (runtime monitor). The principle of RTM is to preset some safe input and output attribute thresholds during the design process and perform real-time monitoring while the NoC is running. When it is found that the input or output value exceeds the set threshold or differs significantly from the set threshold, it means that the NoC is infected by HT, which will remind the designer to look for HT.

C. Related work

DoS attacks have been extensively studied in NoC. However, most publications related to HT detection of NoC focus on improving the structure of NoC or proposing better detection methods.

Researchers in many related fields have proposed new structures to detect hardware Trojans. For example, [11] divides the entire framework into two parts. First, ANN (artificial neural network) is used to detect NoC hardware Trojans, and then Bypass Channel is used to detect hardware Trojans. The infected NoCs are isolated; [3] constructs a novel two-level NoC—Custom many-core architecture with the hierarchical router. Then uses feature correlation analysis to select appropriate feature quantities for detection by multiple machine learning methods; The focus of [12]'s work is to detect LS-DoS, which is a Trojan horse that is not easy to be found. He proposed a specific LS-DoS monitor to detect various parameters of NoC. The realization principle of the monitor is to set Threshold equations that are monitored in runtime.

Some researchers use new methods to detect hardware Trojans. For example, [13] applied ML to detect flooding attacks. First, she extracted enough NoC-related attribute parameters and analyzed them for soundness; [14] selected a large number of NoC features and then used a large number of classification machine learning algorithms to detect hardware Trojans in order to achieve the best results, but sometimes there are oversaturation and detection efficiency is not high ;[15] focuses on using the threshold equation to detect HT, but he uses less machine learning and does not use different benchmark programs to test the effect; [16]'s work simulates four types of HT that can be stimulated by encountering specific conditions Hardware Trojans, using a decision tree, SVM and KNN to detect hardware Trojans respectively.

D. Attacks simulation

This project mainly simulated scenarios where condition-based HTs are activated under certain conditions. Attacks on multi-core routers can affect network packet transfer rates, network/processing core availability, and core communication disruptions [2]. Routers can be attacked externally through memory fabric interfaces or dedicated core interfaces or internally through corrupted routing tables, including Traffic Diversion attack, Routing Loop attacks, and Flooding attacks. These three attacks are also known as DoS attacks, in which the attacked routers are unavailable. Each of them is illustrated in Figure 1, from left to right.

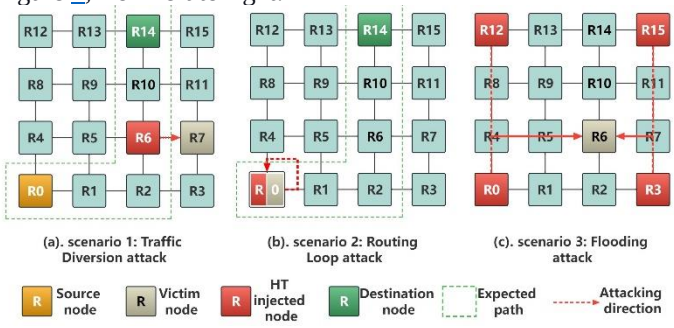

Figure 1 Threat model

a. Traffic Diversion attack (TD)

When the HT-injected router is located in the path of transmission, the destination address of the packets will be changed. The source router is R0, and the destination is R14. When R6 is attacked since R6 is on the transmission path from

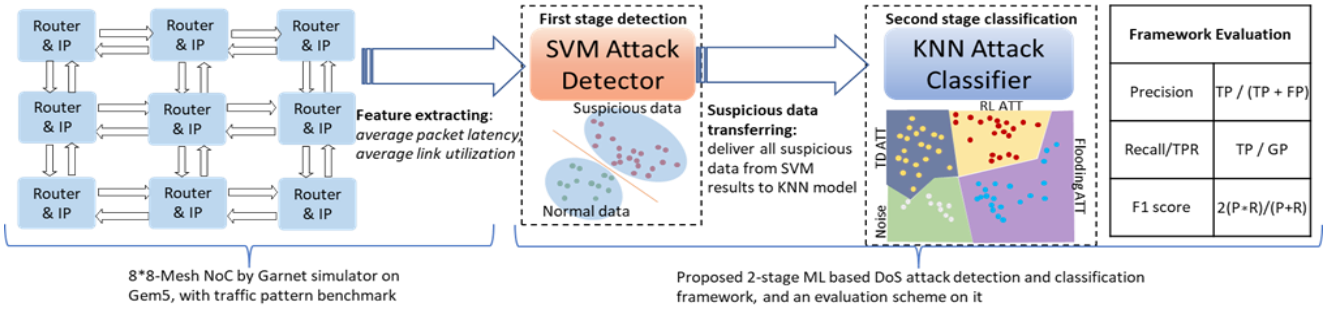

Figure 2 Experiment flow and proposed framework

R0 to R14, the destination will no longer be the original R14, and the data will move to R10.

b. Routing Loop attack (RL)

The attacked router will send the packet back to the core of the node. It will lose the ability to packet transmission between other nodes. As shown in Figure 1, when R0 is attacked, all packets injected from R0 will be returned to the core of R0.

c. Flooding attack

One or more nodes in the NoC could be attacked and they will be designated to transmit packets to a designated node, resulting in redundant packets in the attacked path, causing congestion and packet loss. As shown in Figure 1, when R0, R3, R12, and R15 are attacked, their transmission destination will no longer be random but will be uniformly transmitted to the designated R6.

The dangers of DoS are not only that. It will also change the average hops, packet latency, link utilization, buffer utilization. And more importantly, packet number will dramatically vary due to DoS attack. However, because of its inconstant and unstable variation, the fixed or manually controlled thresholds cannot dynamically and swiftly fit the variation of the attacked system features. Thus, ML based monitoring and detecting schemes are being increasingly employed in NoC security.

III. PROPOSED DOS ATTACK DETECTION AND CLASSIFICATION SCHEME

We first optimized feature selection by correlation analysis to avoid unreliable subjective judgment on features. After that, the proposed 2-stage ML model detection and classification framework were implemented, in which training and validation datasets include all data of normal and three DoS attacks for a more real HT attack simulation.

A. Correlation analysis and features selection

Selecting appropriate attributes based on hardware performance analysis is the first step in developing supervisor ML models. We choose some essential NoC attributes from gem5, which are packets injected, packets received, average packet queueing latency, average packet network latency, average packet latency, average link utilization, average hops, and average power. A Pearson correlation coefficient was used to perform a heatmap for analysis. Figure 3 is the Pearson correlation heatmap on all attributes. The correlation between Packets injected and Packets received is relatively high. Average packet queueing latency, Average packet network latency, and Average packet latency is highly correlated, and we thus selected Average Packet latency as an attribute. Finally, Average link utilization is highly correlated with Average hops. So, we choose Average link utilization as an attribute. At the same time, because of the strong correlation between Average link utilization, Packets injected, and Packets received, in other words, the former can represent others. The last is power. Its correlation with other features is not apparent, so we only analyze it as an NoC performance indicator and not suitable for ML training and prediction. As a result, the attributes we selected are Average packet latency and Average link utilization.

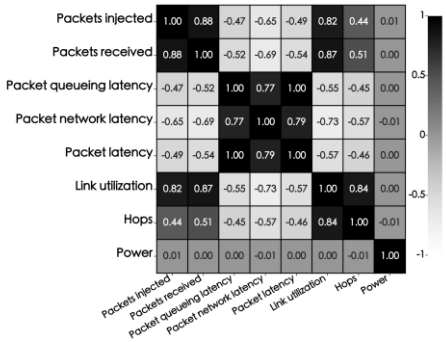

Figure 3 Pearson Correlation heatmap

B. 2-stage ML model detection and classification framework

An overview of the proposed detection & classification framework is illustrated in Figure 2. We tried a variety of supervised ML algorithms: SVM, Linear Regression, KNN, Logistic Regression, Decision Trees, and Random Forests to be each of our proposed two stages respectively. Finally, SVM and KNN show a better balance between execution time, model size, and accuracy.

In the first stage, we trained SVM as the model for attack detection. We need to identify the presence or absence of attacks on the attacked attributes and normal attributes obtained from the simulated NoC. For this typical binary classification problem, SVM is more suitable than others. In addition, [3] presents an expected performance on DoS attacks detection using SVM. I set the proportion of test samples as 25%, which ensured the objective training effect, and set the relaxation variable as 0.5, which ensured the high accuracy and allowed fault tolerance. For the second stage of our framework, we selected KNN as the attacks classification algorithm to classify the suspicious data selected in the first stage into three kinds of attacks, which are more suitable for clustering tasks.

A cascade structure can be observed in Figure 2. We built an 8*8 mesh NoC using Garnet in gem5, tried four traffic

pattern benchmarks, and simulated three main-streaming DoS attacks. When we get the attributes selected, we import them to the SVM attacks detector, which will classify them into normal data and suspicious data. Then the suspicious data predicted by the SVM attack detector will be imported into the KNN attack classifier so that three types of HT-affected data and noise data will be grouped. Finally, the framework performance evaluation index can be obtained, such as precision, recall, etc.

C. *Dataset preparation and model training*

As can be seen in Figure 2, during the beginning of dataset preparation, the runtime Average packet latency and Average link utilization features of different traffic patterns were extracted. After normalization, the impact of data dimensions has been reduced. We established the training and validation datasets which have 2 labels named normal and attacked respectively for the SVM detector, and have 4 labels of normal, TD, RL, and Flooding respectively for the KNN classifier. A synthetically mixed dataset was built to enhance simulation reality and reduce assumptions. Our framework was finally validated on the mixed dataset consisting of HT-infected and normal data under all benchmarks (a total of 16000 data), as there will be the possibility of any attacks occurring at any time. The validation results of the proposed framework were evaluated by a unified and formal performance matrix.

IV. EXPERIMENT RESULTS AND EVALUATION

As Figure 2 shows, We use Gem5 to construct an 8*8 mesh NoC on the plug-in Garnet simulator and run the traffic pattern benchmark at 1GHz. We use three evaluation metrics : Precision, Recall and F1 score. Among them, Precision is TP (True positives) divided by (TP+FP (False positives)), which means that the correct prediction is positive, accounting for the proportion of all positive predictions; Recall is TP divided by (TP+FN (False negatives)), which represents the proportion of correct positive predictions, accounting for all actual positives. The F1 score is twice (precision * recall) divided by (precision + recall). It considers both precision and recalls, let both reach the highest value and take a balanced value. The higher the F1 score, the more robust the model.

As the left figure of Figure 4, on the performance evaluation of the SVM detector, all achieved precisions of around 85%~91%. At the same time, the recall value also fluctuated between 85% and 88%, which means there is a good balance between precision and recall. It also simultaneously breaks out of the constraints of high precision and recall on one ML model. The right figure of Figure 4 is the performance evaluation metrics for the KNN classifier. The precision and recall of the classifier are predictably high and well-balanced in all cases. Uniform_random is of particular attention, as it best fits the characteristics of an actual NoC sending random packets, triggering all three types of attacks. The KNN classifier performs best in the case of uniform_random, obtaining both an expected precision and a superior recall.

Figure 5 shows the precision of the three attacks under different benchmark cases. It can be observed that our trained KNN model has different results for different traffic patterns.

For example, the highest precision of RL attack in Uniform_random is 100% but around 65% in Tornado. This result indicates that the ML models have different adaptations in different application cases. However, the trained ML models are strongly applicable to the uniform_random and all-case mixed datasets, obtaining high detection accuracies.

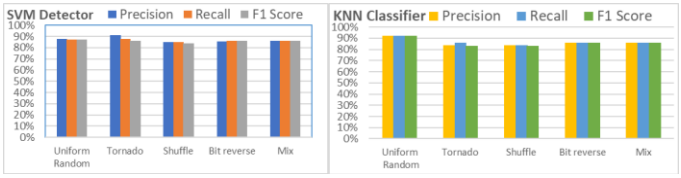

Figure 4 SVM detector and KNN classifier result

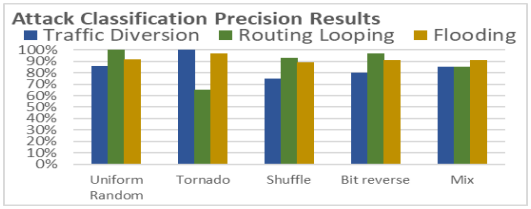

Figure 5 Classification precision of three DoS attacks

Table 1 is a comparison between other related works with our work. For method usage, we combined 2 ML algorithms in cascade, compared to other single ML model techniques. We established a mixed dataset with all attacks rather than a separate single attack. Our proposed scheme achieved 91% precision in the detection phase and at the highest 100% precision in classification on RL and TD attacks, average precision of 93% on flooding attacks.

TABLE I. RESULTS COMPARISON

Works	[3]	[13]	[14]	[Our work]
DoS Attack	TD, RL, Spoofing	Flooding	Flooding	**Flooding, TD, RL**
ML model	SVM	ANN	XGB	**SVM + KNN**
Dataset	Split	Integrated	Integrated	**Mixed**
Detection & Classification	SVM for classification	Sanity check and ANN for classification	XGB for classification	**SVM for detection + KNN for classification**
Precision	94% on Spoofing	89% on Flooding	~96% on Flooding	**Up to 91% detection** **89%~97% on Flooding**
	95% on RL	N/A	N/A	**Up to 100% on RL**
	97% on TD	N/A	N/A	**Up to 100% on TD**

V. CONCLUSION

In this paper, we proposed a framework based on 2-stage ML for DoS attacks detection and classification in NoC. We use four different traffic pattern benchmark programs to model three DoS attacks in different application scenarios and rely on our method to detect and predict attacks in runtime. The HT detector combines high precision and recall, and the ML model is robust. The detection of attack types in a separate benchmark has achieved a very high recognition rate of 90%~100%. The attack detection of all mixed data is also balanced with precision and recall, indicating that our work can detect and classify HT with high accuracy in various unknown scenarios. Its hardware implementation has also been discussed. For example, thanks to the lightweight model of SVM and KNN, its runtime detection and classification speed and power consumption should be accepted based on our estimation. It will be evaluated on the hardware in our future work.

References

[1] Milina, Velichka. "Security in a communications society: opportunities and challenges." *Connections* 11.2 (2012): 53-66.

[2] Gebali, Fayez, Haytham Elmiligi, and Mohamed Watheq El-Kharashi, eds. *Networks-on-chips: theory and practice*. CRC press, 2011.

[3] Kulkarni, Amey, et al. "Real-time anomaly detection framework for many-core router through machine-learning techniques." *ACM JETC* 13.1 (2016): 1-22.

[4] Fang, Dabin, et al. "Robustness analysis of mesh-based network-on-chip architecture under flooding-based denial of service attacks." *2013 IEEE NAS*. IEEE, 2013.

[5] Khattri, Hareesh, Narasimha Kumar V. Mangipudi, and Salvador Mandujano. "Hsdl: A security development lifecycle for hardware technologies." *2012 IEEE HOST*. IEEE, 2012.

[6] Salmani, Hassan. "COTD: Reference-free hardware trojan detection and recovery based on controllability and observability in gate-level netlist." *IEEE TIFTs* 12.2 (2016): 338-350.

[7] Vita, Alessio, et al. "Comparative life cycle assessment of low-pressure RTM, compression RTM and high-pressure RTM manufacturing processes to produce CFRP car hoods." *Procedia CIRP* 80 (2019): 352-357.

[8] Agarwal, Niket, et al. "GARNET: A detailed on-chip network model inside a full-system simulator." *2009 IEEE ISPASS*. IEEE, 2009.

[9] Yu, Qiaoyan, and Jonathan Frey. "Exploiting error control approaches for hardware trojans on network-on-chip links." *2013 IEEE DFTS*. IEEE, 2013.

[10] Ancajas, Dean Michael, Koushik Chakraborty, and Sanghamitra Roy. "Fort-NoCs: Mitigating the threat of a compromised NoC." *DAC*. 2014.

[11] Wang, Ke, Hao Zheng, and Ahmed Louri. "TSA-NoC: Learning-based threat detection and mitigation for secure network-on-chip architecture." *IEEE Micro* 40.5 (2020): 56-63.

[12] Chaves, Cesar G., et al. "Detecting and Mitigating Low-and-Slow DoS Attacks in NoC-based MPSoCs." *ReCoSoC*. IEEE, 2019.

[13] Sinha, Mitali, et al. "Securing an accelerator-rich system from flooding-based denial-of-service attacks." *TETC*: 855-869.

[14] Sudusinghe, Chamika, Subodha Charles, and Prabhat Mishra. "Denial-of-service attack detection using machine learning in network-on-chip architectures." *Proceedings of the 15th IEEE/ACM NOCs*. 2021.

[15] Charles, Subodha, Yangdi Lyu, and Prabhat Mishra. "Real-time detection and localization of DoS attacks in NoC based SoCs." *DATE*. IEEE, 2019.

[16] Hazra, Suvadip, et al. "Evaluation and detection of hardware Trojan for real-time many-core systems." *ISED*. IEEE, 2018.