Overview on deep learning hardware attacks

Maath.F.ismaeel[a, *], Maytham.M.Hammood[b], Qutaiba Alasad[c]

[a]Computer Science Department/ Computer Science and Mathematics College/Tikrit University /Tikrit, Iraq, Email: maath.f.ismaeel@tu.edu.iq

[b]Computer Science Department/ Computer Science and Mathematics College/Tikrit University /Tikrit, Iraq Email: maythamhammood@tu.edu.iq,

[c]Petroleum Systems Control Engineering Department /Petroleum Processing Engineering College/Tikrit University /Tikrit, Iraq, Email: qutaibaeng@tu.edu.iq

Abstract :

Neural networks (NNs) are vital for diverse applications like speech/image recognition, cyber defense, decision-making, and financial forecasting. Deep neural networks (DNNs) are popular for their accuracy, but they demand high memory, power, and complexity.

Hardware attacks (e.g., piracy, Trojans) threaten deep learning platforms. Securing hardware is crucial to safeguarding software integrity. This paper reviews the attacks on deep learning devices and accelerators, briefly reviewing their risks and consequences. Specific attacks (backdoor insertion, model extraction, spoofing, etc.) are addressed through secure hardware design, memory protection, code integrity checks, and secure boot processes. Hardware accelerators for DNNs enhance performance but face security risks. Ensuring confidentiality and integrity against potential threats is crucial. Hardware Trojans (HTs) and other attacks (fault injection, reverse engineering, side-channel) are significant concerns. Effective strategies are necessary to mitigate these threats. Continued research and collaboration among hardware designers, AI/ML researchers, and cybersecurity experts are vital to build a secure foundation for global IC design flows. This enables safe deployment of AI/ML technologies in real-world scenarios.

1. Introduction

Neural networks (NNs) have become a predominant choice for performing tasks, such as speech and image recognition in contemporary research and applications [1−3], defense methodology against cyber-attacks and malware [4, 5], autonomous decision-making systems [6, 7], and modeling high-dimensional distributions [7, 8]. Furthermore, NNs have been proven valuable in diverse domains, including finance, where they are utilized for financial forecasting, risk assessment, and algorithmic trading [9].

Deep neural networks (DNNs) are frequently utilized in a variety of applications, including but not limited to object detection, semantic segmentation, object recognition, and medical imaging, as a result of the expanding success and ongoing development of deep learning technologies [6]. Currently, DNNs can achieve accuracy levels that are higher than those of humans in several applications. Due to the enormous number of network parameters, the system performance is increased significantly, but at the expense of high memory usage, power consumption, and computational complexity.

Hardware attacks are not limited to the malicious user. Many kinds of serious hardware attacks have been introduced last decade. Examples of such severe attacks are intellectual property (IP) piracy, reverse engineering, integrated circuit (IC) counterfeiting, and IC overbuilding [10]. Unlike software attacks, hardware attacks are hard to be prevented or detected. Hardware security is a technique to protect the physical hardware device that the software is installed on it. As a consequence, securing the hardware becomes mandatory and is not less important than securing the software because if the hardware is not trustworthy, then the software is also not trustworthy [11]. This dissertation will concentrate on proposing techniques that can thwart serious hardware attacks with low overhead and finally make the globalization of IC design flow secure against various serious attacks.

For instance, hardware IPs obtained from untrusted vendors may contain malicious implants, backdoors, or other integrity issues, such as hardware Trojans or information leakage. Since the security guarantees of application software depend on the hardware root-of-trust, it is crucial to ensure that the underlying DNN hardawres is free from any security vulnerabilities. Hardware vulnerabilities not only impact the security of the system, but also the overall system reliability. Therefore, it is essential to address these vulnerabilities before deploying the system.

2. Background and Motivation

Hardware security research for deep learning platforms is indeed a rapidly evolving field, with continuous updates and improvements in GPU and TPU architectures to enhance performance and power efficiency. This dynamic environment poses challenges for securing these platforms against hardware attacks. The specific goals of an attacker targeting the physical layer weaknesses in a Deep Neural Network (DNN) are as follows[12]:

- Backdoor Insertion: In this attack, the attacker alters the model stored in memory so that it fails on a subset of tasks while performing correctly on other inputs. The attacker can trigger this failure by providing specific input patterns that activate the backdoor.
- Model Extraction: The attacker tries to extract the model from the device during runtime or by accessing non-volatile memory, such as flash storage. By acquiring the model, the attacker can reverse-engineer it or use it for malicious purposes.
- Spoofing: In a spoofing attack, the attacker corrupts the input data by modifying the environment or tampering with the input sensors. This can lead to the model making incorrect predictions or decisions
- Model Corruption: The attacker aims to compromise the model parameters stored in memory, leading to a failure in all tasks performed by the model. This can result in incorrect outputs and severe consequences if the model is used in critical applications like autonomous vehicles or medical diagnosis.
- Information Extraction: The attacker aims to infer model information from physical side-channels. Side-channel attacks exploit unintended leakage of

information (e.g., power consumption, electromagnetic emissions) to gain insights into the model's behavior or internal structure.

- Sybil Attack: In a collaborative learning platform, the attacker introduces fake devices that participate in the training process. By submitting malicious data or model updates, the attacker aims to manipulate the collaborative learning process and generate invalid or compromised models.

Addressing these hardware-based attacks requires a combination of security measures, including but not limited to:

- Secure hardware design: Incorporating security features directly into the hardware design can mitigate certain types of attacks.
- Memory protection: Employing techniques to protect model parameters stored in memory from unauthorized access or modification.
- Code integrity: Ensuring that the model code remains unaltered during execution.
- Side-channel countermeasures: Implementing techniques to reduce or eliminate information leakage through side channels.
- Input validation: Verifying the integrity and authenticity of input data to detect and prevent spoofing attacks.
- Secure boot process: Ensuring that the system starts up with verified and trusted components.
- Continued research in hardware security is crucial to stay ahead of evolving attack techniques and safeguard deep learning platforms, especially as AI/ML applications become more prevalent in critical domains.

3. Hardware Acceleration on DNNs

In order to alleviate the bottleneck between the DNN's enormous computational load and its training (or inference) pace, a DNN accelerator was been developed. Figure خطأ!

*لا يوجد نص من النمط المعين في المستند.1: Overview of DNN accelerator based on FPGA[13]*the DNN accelerators are specialized domain-specific architectures that have dense parallel computing micro-architectures and a stable memory hierarchy. It can therefore result in increased data throughput bandwidth. As a result, there will be a significant increase in the speed and power efficiency. A well-trained neural network model is frequently mapped into a DNN accelerator for DNN inference acceleration to perform a real-time decision-making. Figure خطأ! لا يوجد نص من النمط المعين في المستند.2:A typical *deep learning accelerator with weights and features broadcasting[14]* The micro-architecture of a DNN accelerator should always be built to be scalable and reconfigurable in order to satisfy the demands of the rapidly evolving network architectures[13].

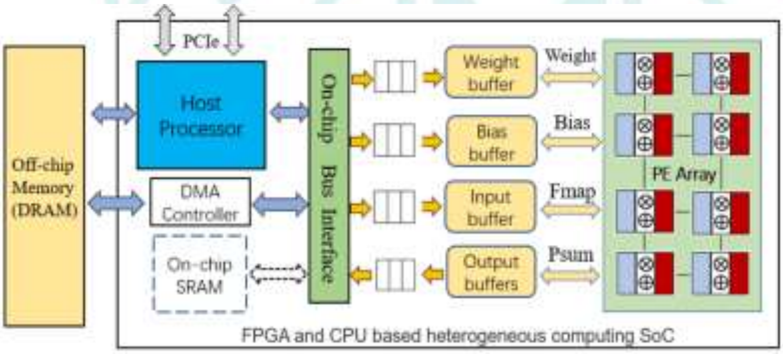

Figure خطأ! لا يوجد نص من النمط المعين في المستند.1: Overview of DNN accelerator based on FPGA[13]

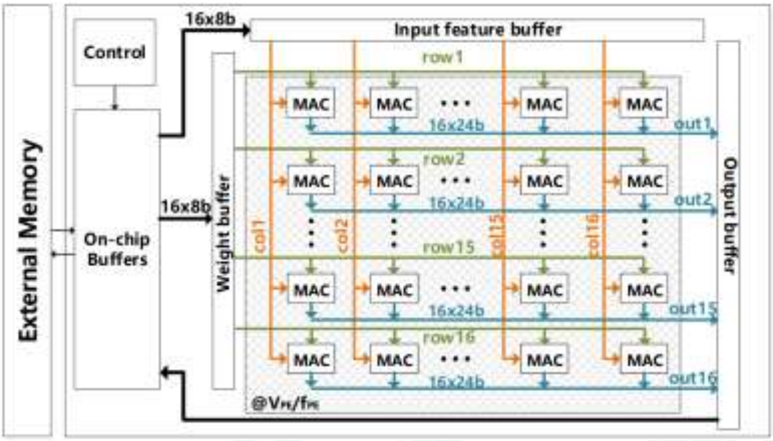

Figure خطأ! لا يوجد نص من النمط المعين في المستند.2:A typical deep learning accelerator with weights and features broadcasting[14]

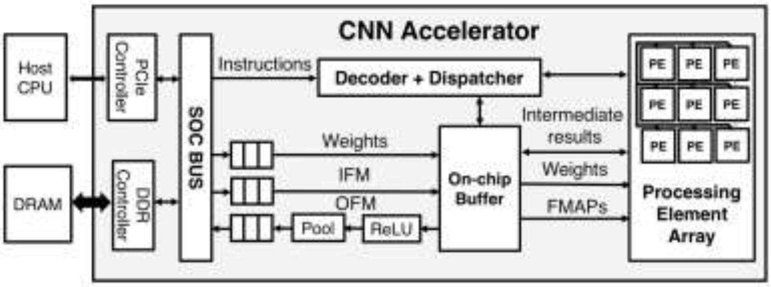

Figure خطأ! لا يوجد نص من النمط المعين في المستند.3: A Typical CNN inference accelerator[15]

Although numerous original ideas for DNN [13-15] accelerators have been developed , they all have the same objective of speeding up the same inference process in DNNs. Figure خطأ! لا يوجد نص من النمط المعين في *المستند.3: A Typical CNN inference accelerator[15]* Practically, all DNN accelerators contain such characteristics as the following:

1- Computation of layers by layers: Since the DNN's layers typically depend on one another in a sequential order, many DNN accelerators compute layer by layer.

2- Multiprocessor computing: Each layer of DNN contains numerous separate computations that can be performed in parallel. the DNN accelerators take the advantage of this characteristic by mapping these individual computations to various processing elements (PEs), which compute the outcomes concurrently.

3- Reusing the data: In the CONV layers, the operands of the computation must be retrieved several times throughout the computation. The DNN accelerators take the advantage of this characteristic by using on-chip memories to facilitate data reuse and lower the number of accesses to off-chip memory.

3.1 Malicious implants (Hardware Trojan)

Hardware Trojan (HT) [16, 17] refers to a form of malicious manipulation carried out on the hardware components. This unauthorized modification has the potential to compromise the security of a system by enabling the leakage of confidential information. Additionally, HTs can adversely impact the performance of the system, leading to a degradation in its overall functionality. Furthermore, these malicious modifications have the capability to instigate the denial-of-service attacks, impeding the normal operation of the affected system. There are two crucial components to this malicious modification of the target integrated circuit (IC): the trigger and payload. The payload makes the harmful behavior possible when the trigger is engaged [18]. For instance, when the trigger logic's output is true, the payload XOR gate's output will invert the desired result. To remain concealed during the regular execution, the trigger is often built by utilizing a collection of uncommon occurrences, such as uncommon signals or uncommon transitions. The payload is the malicious effect that HT will have on the target design, which typically leads to information leaking or incorrect execution. HT may be employed at many levels, such as the bus-level and the IP-level [19]. Because Trojans are typically covert and can only be activated under very specific circumstances, their identification them can be extremely difficult [16]. Because HT is so covert, it is impossible to identify them using the conventional functional validation techniques [20].

3.2 Fault Injection Attacks

Fault injection has the objective of exploiting vulnerabilities by intentionally introducing faults or errors into a system, to compromise its security or gain unauthorized access [21]. Attackers manipulate the hardware or software of the target system to inject faults, which can involve tampering with the power supply, clock signals, voltage levels, or introducing malicious inputs. The impact of fault injection attacks can be significant, leading to system failures, bypassing the security mechanisms, extraction of sensitive information, or disruption of normal system operation [22]. Two of the most common error injection attacks are the laser beam and row hammer, which are used to inject errors into memory and they can change logical values into memory with high accuracy [23].

3.3 Reverse Engineering

Reverse engineering of DLAs [24], is an invasion technique that is accomplished by taking advantage of the backdoors in a DLA, which is either created accidentally or intentionally [25]. The attacker can disclose the IC design and obtain its gate−level net list to further deduce its functioning through a successful reverse engineering attack [26−28]. As a result, internal design information will become known, allowing attackers to illegally copy ICs for their own reasons [29].

3.4 Side−Channel Attacks (SCA)

Electronic equipment inherently emits physical emanation during operation, for instance, but not limited to, execution time and power usage, route delay, and electromagnetic emanation, which gives rise to side− channel vulnerabilities [30]. The hardware devices' private information may mistakenly be revealed by their physical traces. A common method of abusing the side−channel vulnerability, for instance, is through a timing attack, which makes use of timing information to reveal memory information. Assume that the attacker wants to know the precise index of a private array in memory that has one pre−recorded entry in the cache. This can be done by iterating over the full array and measuring access times. the spot with the shortest access times should be the target location due to the significant disparity in access times between cache and memory. The infamous Spectre and Meltdown attacks both heavily utilize the cache-based side-channel attack [31].

4. Conclusion

Neural networks and deep learning technologies have become indispensable tools in various domains, including image and speech recognition, cybersecurity, finance, and autonomous decision-making systems. With the continuous advancements in hardware architectures, specialized DNN accelerators have been developed to handle the computational load of deep learning tasks efficiently, resulting in increased speed and power efficiency.

However, the increasing reliance on deep learning platforms also opens up new avenues for hardware-based attacks, which pose serious threats to the security and reliability of the systems. These attacks include backdoor insertion, model extraction, spoofing, model corruption, information extraction through side-channels, and sybil attacks. Protecting against such hardware attacks is essential, as they can compromise the confidentiality, integrity, and availability of critical applications.

To secure hardware against potential attacks, various techniques need to be employed, including secure hardware design, memory protection, code integrity checks, side-channel countermeasures, input validation, and secure boot processes. Addressing these vulnerabilities becomes even more critical as AI/ML applications become widespread in domains where security and safety are of utmost importance.

Among the specific threats discussed, hardware Trojans (HTs) are particularly concerning, as they can be surreptitiously inserted into integrated circuits, compromising system security and performance. Fault injection attacks, reverse engineering, and side-channel attacks also present significant risks, and effective mitigation strategies are required to thwart these threats.

Ongoing research and development in hardware security for deep learning platforms are imperative to keep up with the evolving landscape of hardware attacks. Collaboration between hardware designers, AI/ML researchers, and cybersecurity experts is essential to ensure the trustworthiness of deep learning platforms and safeguard critical applications from potential threats. By investing in robust security measures and continually updating hardware security practices, we can work towards building a secure foundation for the globalization of IC design flows, enabling the safe deployment of AI/ML technologies in various real-world scenarios.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM,* vol. 60, no. 6, pp. 84-90, 2017.

[2] T. Mikolov, A. Deoras, D. Povey, L. Burget, and J. Černocký, "Strategies for training large scale neural network language models," in *2011 IEEE Workshop on Automatic Speech Recognition & Understanding*, 2011: IEEE, pp. 196-201.

[3] G. Hinton *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal processing magazine,* vol. 29, no. 6, pp. 82-97, 2012.

[4] N. Fatehi, Q. Alasad, and M. Alawad, "Towards Adversarial Attacks for Clinical Document Classification," *Electronics,* vol. 12, no. 1, p. 129, 2023.

[5] Q. Alasad, M. M. Hammood, and S. Alahmed, "Performance and Complexity Tradeoffs of Feature Selection on Intrusion Detection System-Based Neural Network Classification with High-Dimensional Dataset," in *International Conference on Emerging Technologies and Intelligent Systems*, 2022: Springer, pp. 533-542.

[6] P. Abolghasemi, A. Mazaheri, M. Shah, and L. Boloni, "Pay attention!-robustifying a deep visuomotor policy through task-focused visual attention," in

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4254-4262.

[7] N. Karim, A. Zaeemzadeh, and N. Rahnavard, "RL-Ncs: Reinforcement learning based data-driven approach for nonuniform compressed sensing," in *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*, 2019: IEEE, pp. 1-6.

[8] M. Edraki, N. Rahnavard, and M. Shah, "Subspace capsule network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, vol. 34, no. 07, pp. 10745-10753.

[9] M. Elhoseny, N. Metawa, G. Sztano, and I. M. El-Hasnony, "Deep learning-based model for financial distress prediction," *Annals of Operations Research,* pp. 1-23, 2022.

[10] M. Yasin and O. Sinanoglu, "Transforming between logic locking and IC camouflaging," in *2015 10th International Design & Test Symposium (IDT)*, 2015: IEEE, pp. 1-4.

[11] M. Rostami, F. Koushanfar, and R. Karri, "A primer on hardware security: Models, methods, and metrics," *Proceedings of the IEEE,* vol. 102, no. 8, pp. 1283-1295, 2014.

[12] Q. Xu, M. T. Arafin, and G. Qu, "Security of neural networks from hardware perspective: A survey and beyond," in *Proceedings of the 26th Asia and South Pacific Design Automation Conference*, 2021, pp. 449-454.

[13] P. Li and R. Hou, "Int-Monitor: a model triggered hardware trojan in deep learning accelerators," *The Journal of Supercomputing,* vol. 79, no. 3, pp. 3095-3111, 2023.

[14] W. Liu, "Fault-injection based attacks and countermeasure on deep neural network accelerators," 2021.

[15] W. Hua, Z. Zhang, and G. E. Suh, "Reverse engineering convolutional neural networks through side-channel information leaks," in *Proceedings of the 55th Annual Design Automation Conference*, 2018, pp. 1-6.

[16] K. Xiao, D. Forte, Y. Jin, R. Karri, S. Bhunia, and M. Tehranipoor, "Hardware trojans: Lessons learned after one decade of research," *ACM Transactions on Design Automation of Electronic Systems (TODAES),* vol. 22, no. 1, pp. 1-23, 2016.

[17] Q. Alasad, J.-S. Yuan, and Y. Bi, "Logic locking using hybrid CMOS and emerging SiNW FETs," *Electronics,* vol. 6, no. 3, p. 69, 2017.

[18] J. Francq and F. Frick, "Introduction to hardware Trojan detection methods," in *2015 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2015: IEEE, pp. 770-775.

[19] Z. Pan and P. Mishra, "A survey on hardware vulnerability analysis using machine learning," *IEEE Access,* vol. 10, pp. 49508-49527, 2022.

[20] X. Wang, Y. Zheng, A. Basak, and S. Bhunia, "IIPS: Infrastructure IP for secure SoC design," *IEEE Transactions on Computers,* vol. 64, no. 8, pp. 2226-2238, 2014.

[21] S. Hong, P. Frigo, Y. Kaya, C. Giuffrida, and T. Dumitraş, "Terminal brain damage: Exposing the graceless degradation in deep neural networks under hardware fault attacks," in *28th USENIX Security Symposium (USENIX Security 19)*, 2019, pp. 497-514.

[22] X. Hou, J. Breier, D. Jap, L. Ma, S. Bhasin, and Y. Liu, "Security evaluation of deep neural network resistance against laser fault injection," in *2020 IEEE International Symposium on the Physical and Failure Analysis of Integrated Circuits (IPFA)*, 2020: IEEE, pp. 1-6.

[23] Y. Liu, L. Wei, B. Luo, and Q. Xu, "Fault injection attack on deep neural network," in *2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2017: IEEE, pp. 131-138.

[24] M. Fyrbiak *et al.*, "Hardware reverse engineering: Overview and open challenges," in *2017 IEEE 2nd International Verification and Security Workshop (IVSW)*, 2017: IEEE, pp. 88-94.

[25] T. Meade, S. Zhang, and Y. Jin, "Netlist reverse engineering for high-level functionality reconstruction," in *2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2016: IEEE, pp. 655-660.

[26] R. Torrance and D. James, "The state-of-the-art in semiconductor reverse engineering," in *Proceedings of the 48th Design Automation Conference*, 2011, pp. 333-338.

[27] Q. Alasad and J. Yuan, "Logic obfuscation against IC reverse engineering attacks using PLGs," in *2017 IEEE International Conference on Computer Design (ICCD)*, 2017: IEEE, pp. 341-344.

[28] Q. Alasad, Y. Bi, and J.-S. Yuan, "E2LEMI: energy-efficient logic encryption using multiplexer insertion," *Electronics,* vol. 6, no. 1, p. 16, 2017.

[29] A. Alaql, S. Chattopadhyay, P. Chakraborty, T. Hoque, and S. Bhunia, "LeGO: A learning-guided obfuscation framework for hardware IP protection," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 41, no. 4, pp. 854-867, 2021.

[30] Q. Alasad, J.-S. Yuan, and P. Subramanyan, "Strong logic obfuscation with low overhead against IC reverse engineering attacks," *ACM Transactions on Design Automation of Electronic Systems (TODAES),* vol. 25, no. 4, pp. 1-31, 2020.

[31] Q. Alasad, J. Lin, J.-S. Yuan, D. Fan, and A. Awad, "Resilient and secure hardware devices using ASL," *ACM Journal on Emerging Technologies in Computing Systems (JETC),* vol. 17, no. 2, pp. 1-26, 2021.