Rethinking FPGA Security in the New Era of Artificial Intelligence

Xiaolin Xu
Department of Electrical and Computer Engineering
University of Illinois at Chicago
Email: xiaolin8@uic.edu

Jiliang Zhang
CSEE, Hunan University & Cyberspace Security
Research Center, Peng Cheng Laboratory, China
Email: zhangjiliang@hnu.edu.cn

***Abstract*—With various possible attacks against commercial electronic devices reported over the past few decades, the security of hardware devices and systems has become an urgent problem. Accordingly, a large number of solutions and countermeasures have been explored to mitigate these attacks. Artificial intelligence, as one of the fastest-growing research areas, also makes a unique impact on the landscape of vulnerabilities and countermeasures of hardware. As a vital subset of artificial intelligence, machine learning algorithms are found of great use in hardware security from both constructive and destructive perspectives. In this paper, we provide a survey of such *double-edged sword* impact of machine learning techniques on the security of hardware. In particular, we focus on the discussion of FPGA security. We enumerate both countermeasures and attacks based on pure machine learning algorithms, as well as the integration of machine learning and other methods, such as side-channel analysis. In addition, we also discuss the security concerns of FPGAs when they are used as carriers or accelerators for machine learning algorithms. Specifically, we present the security issues of FPGAs in two different application scenarios: 1) as a standalone computing resource and 2) as a public-leased computing resource shared by multiple users.**

I. Introduction

The past decades have witnessed significant development of the semiconductor industry. All modern critical infrastructures such as power grids, public transportation systems, and even national defense systems are built on numerous integrated circuits (ICs). As a result, the security, quality, and assurance of the critical infrastructures are closely relying on the trustworthiness of the underlying ICs, which is being threatened for the following reasons: 1) many emerging hardware technologies have been embedded into computing resources for acceleration purpose, without considering the design rules for security; 2) due to the improper security practices, incomplete security checks, and faster production cycles, a hardware vulnerability may appear at a previously-ignored point; 3) the vast deployment of mobile devices, sensors, and actuators makes it possible for a small set of malicious devices to devastate commerce, government operation, and even national defense to a great extent; 4) the globalization of the semiconductor foundry business poses risks from untrusted fabrication and distribution, where Trojans insertion, IP cloning, and counterfeits may happen. With different hardware vulnerabilities are being explored and reported, the trustworthy of hardware devices are drawing significant attention. Meanwhile, we found that the development and deployment of various machine learning (ML) algorithms have also played an important role in hardware security. For example, the wide application of ML techniques has brought impact to the hardware security area, including novel solutions for some challenging problems, as well as some unique threat models and vulnerabilities.

To help researchers better understand the challenges and opportunities that fall into the intersection between hardware security and ML, in this paper, we summarize the past, ongoing, and predict potential future works that relate to these two topics. Our survey mainly focuses on three aspects: (1) The destructive and constructive impacts on hardware security (especially on FPGA security) of ML techniques. For example, ML has been widely used to attack some well-known hardware security primitives, such as physical unclonable functions. Meanwhile, ML techniques are also being constructively integrated with side-channel analysis to mitigate hardware vulnerabilities (e.g., hardware Trojan). (2) The security of hardware devices and platforms are broadly used as accelerators for various ML algorithms. In this paper, we focus on discussing the security of FPGAs, which are being used as one of the major platforms to speed up ML applications. Specifically, we survey and discuss the security issues of FPGA in two different scenarios: 1) FPGA is used as a single node accelerator for ML algorithms, and 2) FPGA is integrated into cloud-service and used by multiple users for computing-intensive applications.

The remainder of this paper is organized as follows: Section II discusses the application of ML techniques in conventional hardware security. Specifically, we present the use of ML techniques in attacking hardware security primitives and its useful application in help with detecting hardware Trojans. Section III presents the security issues associated with FPGA as the hardware accelerator for ML algorithms. Section IV concludes this paper with the discussion of future works.

II. The Impacts of Machine Learning on FPGA Security

A. ML Attacks on FPGA Security Primitives

On-chip key storage and generation have been used to authenticate different hardware devices. As an important hardware platform, the authentication and identification of FPGAs are challenging problems due to the lack of nonvolatile

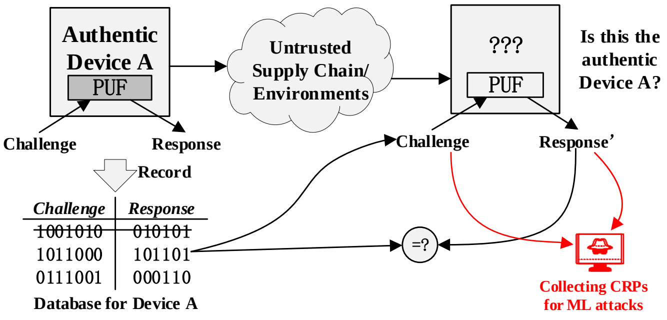

Fig. 1. ML attacks on PUFs

memory. To mitigate this limit, physical unclonable functions (PUFs) are proposed as a key technique to generate and store digital keys on FPGAs. PUF are promising hardware security primitive for lightweight authentication, as shown in Fig. 1. A large number of FPGA PUF structures have been proposed over the last decades. However, ML has become a major threat to most of these PUFs [12]. Currently, ML attacks on PUFs can be roughly classed into pure ML attacks and hybrid attacks.

1) Pure ML attacks. For a PUF structure, ML techniques can learn its complex input-output mapping from a small number of challenge and response pairs (CRPs) to make accurate predictions of the unknown responses (see Fig. 1). However, the computing complexity of ML modeling attacks increases exponentially with the scale of PUF and the number of nonlinear logical elements in a PUF [2]. In [3] and [59], Rührmair *et al.* used Support Vector Machine (SVM), Logistic Regression (LR), and Evolution Strategy (ES) to attack a series of PUFs. The CRPs are collected from simulation and actual FPGA and ASIC PUF implementations. The experimental results show that the behavior of these PUFs can be accurately modeled with ML algorithms. Recently, Shi *et al.* proposed an approximation attack on strong PUFs [4]. It includes logical approximation and global approximation attacks, which are used to model strong PUFs from the local and global aspects, relatively. If the structure of PUF can be decomposed into logic gates, which can be expressed by approximation functions, the logical approximation can be launched. If the PUF structure contains too many logic gates, the global approximation is used as an alternative. To get the global approximation, the PUF structure should be analyzed to extract digital features from the mapping relationship between the input and output. Then a mathematical function with similar characteristics is concluded as the global approximation function, which can be used to build a high-precision ML model for PUFs.

2) Hybrid attacks based on side channel and ML. To reduce the modeling time and enhance the attack ability, hybrid attacks use the auxiliary information from the side channel to assist the ML algorithm in modeling the PUF [5]. A hybrid attack combining Differential Power Analysis (DPA) and LR is proposed in [6], which can distinguish subtle changes by comparing power traces before and after the response to extract the power consumption of the XOR gate implemented on FPGAs. In [16], Becker proposed a hybrid attack combining the Correlation Power Analysis (CPA) and the Covariance Matrix Adaptation Evolution Strategy (CMA-ES), which uses power correlation coefficient as the fitness function to test the applicability of CMA-ES. This method makes it possible to conduct modeling attacks, even in noisy environments, successfully. In [16], Becker proposed a hybrid attack that combines reliability and CMA-ES. The attack utilizes the fact that the response bit is more likely to be flipped if the PUF delay difference is closer to 0. The response of the same challenge is measured in different voltage environments, and then the attacker could use the reliability information's correlation coefficient, which caused by flipping as the fitness function for CMA-ES. In [8], a hybrid attack that combines repeatability and least squares is proposed, in which the attacker found that the repeatability is linear with the challenge in the range of [0.1, 0.9]. They collected the challenge-repeatability pairs in the linear region and used the least-squares to establish the delay model for attacking.

As discussed above, ML attacks have posed a significant threat to strong PUF. Meanwhile, various nonlinear PUF structures or obfuscation techniques that can defend against ML attacks are also being proposed [9].

1) Structural nonlinearization.: This method is to add nonlinear elements to the PUF structure to resist ML attacks, such as XOR Arbiter PUF [10], lightweight PUF [11], feed-forward Arbiter PUF [12], and Multiplexer PUF [13]. However, the improved nonlinearization PUF structure can still be modeled as long as the size and complexity are fixed [14]. More severely, the use of nonlinear elements also dramatically decreases the reliability of PUF.

2) CRP obfuscation.: This solution is to hide the mapping between challenge and response to prevent the attackers from modeling the PUF [15]. But adding some massive obfuscation structures (e.g., hash function) to the PUF structure would incur high hardware overhead. Besides, if the response is directly obfuscated, the hardware overhead of error correction on the responses will be high [16]. Recently, Wang *et al.* [33] proposed to contaminate the training data (obfuscated data) by flipping a portion of the response to inject inconsistent CRPs so that an accurate model of the PUF is challenging to be built by ML algorithms due to the obfuscation. Note that besides modeling attacks, the ML-based algorithms is also used to improve the reliability of PUFs [63].

B. Hardware Trojan Detection based on ML Techniques

In this section, we present other constructive applications of ML-techniques in hardware security. The outsourced fabrication of semiconductor products has witnessed significant improvement over the past decades. The new globalized semiconductor business model and supply chain, although speed up development the semiconductor industry and shorten the time-to-market of electronic devices and products, but also creates new security concerns. Without being monitored, the electronic designs (e.g., the `GDSII` file and the bitstream file for FPGA) might encounter various security issues, such as counterfeit [34], reverse engineering [35], IP cloning [36], and more severely, hardware Trojan insertion [37]. Hardware Trojan, as a hot topic in hardware, has been deeply investigated in the past a few years, to motivate the research on hardware Trojan detection as well as providing a generic standard, a set of validated Trojan benchmarks are shared in Trusthub [41]. A large number of hardware Trojan design [37], detection [39], and avoidance [40] schemes have been proposed and discussed. Among these hardware Trojan detection and prevention methods, ML techniques have found great potential in mitigating hardware Trojans. Specifically, ML techniques have been applied in detecting hardware Trojans in two perspectives: 1) Accelerating hardware Trojan detection with ML-based classifier, and 2) Detecting hardware Trojan with the combined scanning electron microscope (SEM) imaging and ML-based computer vision techniques.

1) Conventional Trojan detection accelerated by ML: Since a hardware Trojan works by violating the originally designated functionalities of a circuit, thus the activation of it would inevitably incur some microscopic difference, such as timing latency [42], power-trace [43], and electromagnetic leakage [44]. With the circuitry becoming more complicated and the Trojan much smaller, the detection techniques based on pure mathematical analysis become less effective. To overcome these shortcomings, ML-based classifiers are used for building labels for different Trojan-detecting features, such as electromagnetic leakage [45]. All these methods train a classifier based on the "should-be" behavior of Trojan-free circuit, thus the well-trained model will identify the Trojan-inserted circuit through its label. However, since all the circuits are fabricated together, it is thus usually impractical to obtain golden chips that can be used for Trojan detection.

2) Computer-vision-based Trojan Detection: Thanks to the advancement of ML-based computer vision and scanning electron microscope (SEM) techniques, the original circuit designer can use the microscopic difference between the layout of authentic and Trojan-inserted circuits for Trojan detection [46]. Specifically, the layout of used for circuit construction are obtained via SEM scanning from the backside of the circuitry, and the image of various standard gates are fed into a supervised ML framework for classification. Since the insertion of Trojan usually inevitably introduced edits on the netlist or layout of the circuit, thus incurring changes at the gate-level layout [47]. Therefore, the modified portion of a circuit can be quickly identified by such computer-vision-based frameworks.

C. IC camouflaging enhanced by ML

IC camouflaging is an active countermeasure against the reverse engineering of outsourced circuit fabrication or FPGA bitstream, which shows effectiveness in mitigating different de-camouflaging attacks, such as the SAT-based attack [56]. However, although a large number of camouflaging methods have been proposed, there still lacks a generic standard to evaluate the complexity of them. As a result, the robustness of existing camouflaging methods is usually over-estimated against some known attacks [38]. More severely, the existing methods are mostly based on the employment of camouflaging cells with large overhead. These weaknesses significantly limit the further development and adoption of these camouflaging methods.

ML-based methods have found great use in solving the problems mentioned above. In [57], Li *el al.* proposed an active learning-based scheme, which can accurately quantify the computing complexity in breaking an IC camouflaging method. Specifically, the SAT-based attacking method is used as the golden sample to train an ML model, through which a new criterion called *de-camouflaging complexity* is generated. This criterion counts the smallest number of required input-output patterns to break a camouflaged circuit, thus stands for the robustness of any camouflaging technique against attacks. In other words, any camouflaged circuit that achieves high de-camouflaging complexity is also secure against these well-known attacks. Thanks to the use of active learning, the large overhead of conventional IC camouflaging cells is also significantly reduced. Two lightweight camouflaging strategies: XOR-type that uses dummy contacts and stuck-at-fault-type that leverages doping-based methods are proposed [57].

III. Security of FPGA as an Accelerator for Machine Learning

Besides the direct use of ML techniques in attack and protecting hardware devices, there also exists another intersection between ML development and hardware systems, that that various hardware devices and platforms (e.g., FPGA and GPU) are broadly used as accelerators for ML algorithms. For brevity, this section focuses on the discussion of FPGA, as one of the major platforms to speed up ML implementations. We consider FPGA in two different application scenarios: 1) as a single-node hardware accelerator and 2) as a public cloud-computing resource leased by multiple users.

A. Adversarial Attacks on FPGA-based DNN Accelerator

Deep neural networks (DNNs) have shown huge superiority over humans in image recognition, speech processing, autonomous vehicles, and medical diagnosis. However, recent studies indicate that DNNs are vulnerable to various attacks, such as adversarial examples [19], physical attacks [20], and backdoor attacks [31]. At the training stage, poisoning attacks [22] can damage the original probability distribution of training data by injecting malicious examples to reduce the prediction accuracy of the model. At the test or inference stage, evasion attacks [23] can trick a target system by constructing a specific input example without changing the target ML system. In 2005, Lowd and Meek [24] proposed the concept of adversarial learning, in which an adversary conducts an attack to minimize a cost function. Under this framework, they proposed an algorithm to reverse engineer linear classifiers. In 2006, Barreno *et al.* [25] presented a taxonomy of different types of attacks on ML systems. To mitigate poisoning attacks and evasion attacks, a lot of defenses have been proposed [26][27]. In 2014, Szegedy *et al.* [28] proposed the concept of *adversarial example*. By adding a slight perturbation to the input, the model misclassifies the adversarial example with high-confidence, while human eyes cannot recognize the difference. Even though different models have different architectures and training data, the same set of AEs can be used to attack the related models. As a result, the adversarial examples have become a big concern to the robustness and security of DNNs.

Dedicated FPGA-based DNN accelerators are expected to gain mainstream adoption in the foreseeable future due to their high-performance computing capabilities and flexibility for reconfiguration [18]. However, the vulnerabilities associated with the hardware platforms also incur serious security concerns. For example, in addition to these attacks by adversarial examples, physical attacks that target the hardware implementations of DNNs are also becoming an emerging threat as well. Current physical attacks on DNNs mainly include side-channel attacks and fault injection attacks. Side-channel attacks can recover the input image from the collected power traces without knowing the detailed parameters in the neural network [29]. Fault injection attacks modify DNN parameters stored in memory by the memory fault injection techniques such as laser beam and row hammer, which can change the classification of certain input images to the target labels [20][32].

A neural network backdoor is a hidden pattern injected during training, which forces the model to generate the same error behavior for input with a specific trigger. It does not affect its classification task on clean samples. With the rise of outsourced training and transfer learning, this kind of backdoor attack brings challenges to the security of the neural network. Recently, Liu *et al.* [30] proposed a Trojan attack on neural networks, which can design triggers from the original model according to the values that cause significant responses of some neurons, to establish a stable relationship between triggers and neurons, and inject active backdoor with fewer training samples. Yao *et al.* [31] proposed hidden backdoors embedded in the teacher model that are retained during the transfer learning and only activated once the model is customized to identify the target label. In recent years, several defenses on backdoor attacks have been put forwarded [32][33].

B. Security of FPGA as a Cloud-service

The application and deployment of ML techniques and algorithms make high-performance computing-capability an urgent need. Correspondingly, reconfigurable computing devices like FPGAs have been integrated into various platforms for performance acceleration, such as the cloud-computing service. Nevertheless, the rapid development of semiconductor design and fabrication technologies also make it possible to have FPGAs with larger size and high complexity. In this scenario, more FPGAs have been deployed by leading cloud service providers like Amazon and Microsoft on multi-tenant servers to provide reconfigurable and high-performance computing capabilities [48] [49]. For example, the Stratix-V FPGAs from Altera has been deployed by Microsoft Azure for domain-specific computing [50]; Amazon released their EC2 F1 instances equipped with programmable hardware (UltraScale+VU9P FPGAs) from Xilinx [51]. Alongside the rapid growth of cloud computing market, it has been predicted that by 2020, as many as 30% data centers will be equipped with FPGAs [52], and around 30x speedup can be achieved for various applications like financial risk analysis, big data search, and cryptographic algorithms [51].

The recent developments of FPGA-based cloud computing imply that it becomes possible to have multi-users reside in the same FPGA chip. This advancement has facilitated the development of high-performance computing but also brings new threats to the security of FPGA and cloud computing. For example, the "co-tenancy" of multiple tenants on an FPGA chip has created a unique attack surface, where the malicious users on the same FPGA chip may have access to the secret information of other users. In practical multi-tenant FPGA environments, many hardware resources are jointly used by several tenants. Therefore, an attacker (malicious tenant) can possibly leverage such *indirect* interaction with other tenants to implement various attacks. These threat models are relatively new and different from the well-studied threat models for FPGA security, thus needs more efforts from the research community.

1) Threat model: With the recent deployment of multi-tenant FPGAs on cloud computing service, several attacks have been proposed [53][54][58][60]. Without loss of generality, in this paper, we follow the threat model in accordance with most of these works [61] [58][60][62], which has the following features: 1) multiple tenants "co-reside" on an FPGA chip independently from each other, and their circuits can be executed at the same time, 2) each tenant has his bitstream and the freedom to implement his designs in desired regions (if not taken by others) on an FPGA chip, and 3) all tenants jointly use the resources on an FPGA chip. For example, power supply through the power distribution network, but there is no physical interaction (e.g., shared logic or circuit) between any two tenants. The threat model is illustrated in Fig. 2, in which three tenants (A, B, and C) co-reside on the user-programmable logic (PL) part of a cloud-FPGA chip, and each tenant has her independent implementations (denoted with *IP core*). Tenant C represents the malicious tenant, and the other two tenants represent the victim ones. Different hardware components (IPs) are implemented with different Configurable Logic Blocks (CLBs) that are connected by long-wires.

2) Side-channel attacks: It has been demonstrated that the capacitive crosstalk between FPGA long-wires can be used as a new side-channel [62] [61]. Capacitive crosstalk is caused by the significantly reduced dimension of integrated circuits. In [61], the secret key of advanced encryption standard (AES) is successfully extracted by such side-channel attacks [61]. This phenomenon has been validated with various commodity FPGAs from Xilinx and Intel/Altera. For example, Xilinx Artix-7 FPGA has two classes of long-wires: vertical long-wire (denoted as VLONGs) and horizontal long-wire (denoted

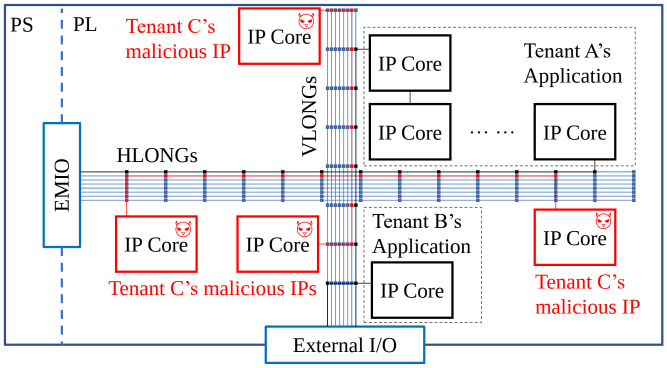

Fig. 2. Threat model for multi-tenant FPGA [55].

as HLONGs), both of which can be used for conducting side-channel attacks. The information leakage caused by capacitive crosstalk between FPGA long-wires is stable against temperature fluctuation and other noise, which makes the defense more challenging.

3) Fault injection attacks: The *co-residency* of multiple tenants on an FPGA chip makes the shared resource (e.g., power distribution network and reconfigurable fabrics) a new medium to launch attacks, which can be exploited for malicious purposes in various ways. In [58] and [60], the real-time power trace of a victim tenant is collected by voltage sensors for power analysis attacks. Specifically, such sensors can be easily crafted with ring oscillator (RO) using the reconfigurable logic on FPGAs. [64] successfully that through manipulating the voltage drop on an FPGA, the secret key of an advanced encryption standard (AES) implementation can be extracted by the malicious user. Similarly, a power attack targeting the entropy of the true random number generator (TRNG) on FPGAs is implemented and demonstrated in [65]. As a result, the entropy of the random numbers of the TRNG circuit is corrupted by the power attacks [66].

IV. Conclusion

In this paper, we briefly introduced the application of machine learning (ML) in hardware security. Although the use of machine learning in attacking hardware security primitives like PUFs has been extensively studied, its constructive applicability for mitigating hardware security issues has not been thoroughly explored. We summarize a few ML-enhanced solutions for hardware Trojan detection and IC camouflaging to demonstrate that ML-techniques also have great potential in mitigating hardware security issues and thus needs more exploration. Besides, we present an under-explored security issue of hardware devices and system, that when they are used as accelerators for ML algorithms. With the ever-increasing complexity of hardware devices and systems to support computing-intensive tasks, we envision that more vulnerabilities will be revealed, for which we also expect that the community will propose more countermeasures.

References

[1] J. W. Lee, D. Lim, B. Gassend, G. E. Suh, M. van Dijk, and S. Devadas, "A technique to build a secret key in integrated circuits for identification and authentication applications", *2004 Symposium on VLSI Circuits. Digest of Technical Papers*, pp. 176-179, 2004.

[2] 1 U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, and J. Schmidhuber, "Modeling attacks on physical unclonable functions", *Proceedings of the 17th ACM Conference on Computer and Communications Security*, pp. 237-249, 2010.

[3] U. Rührmair, and J. Sölter et al. , "PUF Modeling Attacks on Simulated and Silicon Data", *IEEE Transactions on Information Forensics & Security*, vol. 8, no. 11, pp. 1876-1891, 2013.

[4] J. Shi, Y. Lu, and J. Zhang, "Machine Learning Attacks on Multiplexer-based PUFs", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2019, DOI: 10.1109/TCAD.2019.2962115.

[5] X. Xu, and W. Burleson , "Hybrid side-channel/machine-learning attacks on PUFs: A new threat?", *Proceedings of the Conference on Design, Automation & Test in Europe*, pp. 1-6, 2014.

[6] U. Rührmair et al., "Efficient Power and Timing Side Channels for Physical Unclonable Functions", *Cryptographic Hardware and Embedded Systems – CHES 2014*, Springer, Berlin, Heidelberg, vol. 8731, pp. 476-492, 2014.

[7] G.T.Becker, "The Gap Between Promise and Reality: On the Insecurity of XOR Arbiter PUFs", *Cryptographic Hardware and Embedded Systems – CHES 2015*, Springer, Berlin, Heidelberg, vol. 9293, pp. 535-555, 2015.

[8] J. Delvaux, and I. Verbauwhede, "Side channel modeling attacks on 65nm arbiter PUFs exploiting CMOS device noise", *IEEE International Symposium on Hardware-Oriented Security and Trust*, pp. 137-142, 2013.

[9] J. Zhang, and C. Shen, "Set-based Obfuscation for Strong PUFs against Machine Learning Attacks", *arXiv: 1806.02011*, 2018.

[10] D. Lim, J. W. Lee, B. Gassend, G. E. Suh, M. Van Dijk, and S. Devadas, "Extracting secret keys from integrated circuits", *IEEE Transactions on Very Large Scale Integration Systems*, pp. 1200-1205, 2005.

[11] M. Majzoobi, F. Koushanfar, and M. Potkonjak , "Lightweight Secure PUFs", *IEEE/ACM International Conference on Computer-Aided Design*, pp. 670-673, 2008.

[12] J.W. Lee, D.Lim, and B. Gassend et al. , "A technique to build a secret key in integrated circuits for identification and authentication applications", *2004 Symposium on VLSI Circuits. Digest of Technical Papers*, 2004.

[13] D. P. Sahoo, D. Mukhopadhyay, R. S. Chakraborty, and P. H. Nguyen, "A Multiplexer-Based Arbiter PUF Composition with Enhanced Reliability and Security", *IEEE Transactions on Computers*, vol. 67, no. 3, pp. 403-417, 2018.

[14] M. Khalafalla, and C. Gebotys, "PUFs Deep Attacks: Enhanced modeling attacks using deep learning techniques to break the security of double arbiter PUFs", *2019 Design, Automation Test in Europe Conference & Exhibition*, pp. 204-209, 2019.

[15] B. Gassend, M. Van Dijk, D. Clarke, E. Torlak, S. Devadas, and P. Tuyls , "Controlled Physical Random Functions and Applications", *ACM Transactions on Information and System Security*, 2008.

[16] G. T. Becker, "On the Pitfalls of Using Arbiter-PUFs as Building Blocks", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 8, pp. 1295-1307, 2015.

[17] Sying-Jyan Wang, Yu-Shen Chen, and Katherine Shu-Min Li, "Adversarial Attack against Modeling Attack on PUFs", *Proceedings of the 56th Annual Design Automation Conference*, pp. 1-6, 2019.

[18] R. Zhao, W. Song, W. Zhang, T. Xing, J. Lin, M. Srivastava, R. Gupta, and Z. Zhang, "Accelerating Binarized Convolutional Neural Networks with Software-Programmable FPGAs", *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pp. 15-24, 2017.

[19] J. Zhang, and C. Li,"Adversarial Examples: Opportunities and Challenges“, *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1-16, 2019.

[20] P. Zhao, S Wang, Cheng Gongye, Y. Wang, Y. Fei, and X. Lin,"Fault Sneaking Attack: a Stealthy Framework for Misleading Deep Neural Networks", *In Proceedings of The 56th Annual Design Automation Conference 2019*, pp. 2-6, 2019.

[21] Y. Liu, L. Wei, B. Luo, and Q. Xu, "Fault Injection Attack on Deep Neural Network", *2017 IEEE/ACM International Conference on Computer-Aided Design*, pp. 13-16, 2017.

[22] B. Biggio, G. Fumera, F. Roli, and L. Didaci. "Poisoning adaptive biometric systems," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7626, pp. 417-425, 2012.

[23] K. Grosse, N. Papernot, P. Manoharan, M. Backes, and P. McDaniel, "Adversarial Examples for Malware Detection," in *European Symposium on Research in Computer Security*, pp. 62-79, 2017.

[24] D. Lowd and C. Meek, "Adversarial learning," *in Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in Data Mining*, pp. 641-647, 2005.

[25] M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D.Tygar, "Can machine learning be secure? (Invited Talk)," *in Proceedings of ACM Symposium on Information, Computer and Communications Security*, pp. 16-25, 2006.

[26] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognit.*, 2018.

[27] M. Brundage et al. , "The Malicious Use of Artificial Intelligence: Forecasting, Prevention, and Mitigation,", arXiv:1802.07228, 2018.

[28] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna,D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *in Proceedings of International Conference on Learning Representations (ICLR)*, 2014.

[29] L. Wei, B. Luo, Y. Li, Y. Liu and Q. Xu, "I Know What You See: Power Side-Channel Attack on Convolutional Neural Network Accelerators", *in 2018 Annual Computer Security Applications Conference*, pp. 3-7, 2018.

[30] Y. Liu, S. Ma, Y. Aafer et al., "Trojaning Attack on Neural Networks," in Proceedings of the 2018 Network and Distributed System Security Symposium, 2018.

[31] Y. Yao, H. Li, H. Zheng, and B. Y. Zhao, "Latent Backdoor Attacks on Deep Neural Networks," in Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, 2019, pp. 2041–2055.

[32] Y. Liu, Y. Xie, and A. Srivastava, "Neural Trojans," in 2017 IEEE International Conference on Computer Design (ICCD), 2017, pp. 45–48.

[33] B. Wang, Y. Yao, S. Shan et al., "Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks," in 2019 IEEE Symposium on Security and Privacy (SP), 2019, pp. 707–723.

[34] L. Wei, C. Song, Y. Liu, J. Zhang, F. Yuan, and Q. Xu, "Boardpuf: Physical unclonable functions for printed circuit board authentication," in *Computer-Aided Design (ICCAD), 2015 IEEE/ACM International Conference on*. IEEE, 2015, pp. 152–158.

[35] J. Rajendran, M. Sam, O. Sinanoglu, and R. Karri, "Security analysis of integrated circuit camouflaging," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security.*

[36] X. Zhang, N. Tuzzio, and M. Tehranipoor, "Identification of recovered ics using fingerprints from a light-weight on-chip sensor," in *Proceedings of the 49th Annual Design Automation Conference*, ACM, 2012.

[37] L. Lin, M. Kasper, T. Güneysu, C. Paar, and W. Burleson, "Trojan side-channels: lightweight hardware trojans through side-channel engineering," in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2009, pp. 382–395.

[38] El Massad M, Garg S, Tripunitara MV, "Integrated Circuit (IC) Decamouflaging: Reverse Engineering Camouflaged ICs within Minutes," in *2015 Network and Distributed System Security Symposium (NDSS).*

[39] P. Swierczynski, M. Fyrbiak, P. Koppe, and C. Paar, "FPGA Trojans through detecting and weakening of cryptographic primitives," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. accepted/to appear, 2015.

[40] G. T. Becker, M. Kasper, A. Moradi, and C. Paar, "Side-channel based watermarks for integrated circuits," in *2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*. IEEE, 2010, pp. 30–35.

[41] M. Tehranipoor, R. Karri, F. Koushanfar, and M. Potkonjak. (2016) Trusthub. [Online]. Available: https://www.trust-hub.org

[42] Y. Jin and Y. Makris, "Hardware trojan detection using path delay fingerprint," in *2008 IEEE International workshop on hardware-oriented security and trust*. IEEE, 2008, pp. 51–57.

[43] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar, "Trojan detection using ic fingerprinting," in *2007 IEEE Symposium on Security and Privacy (SP'07)*. IEEE, 2007, pp. 296–310.

[44] J. Balasch, B. Gierlichs, and I. Verbauwhede, "Electromagnetic circuit fingerprints for hardware trojan detection," in *2015 IEEE International Symposium on Electromagnetic Compatibility (EMC)*. IEEE.

[45] T. Iwase, Y. Nozaki, M. Yoshikawa, and T. Kumaki, "Detection technique for hardware trojans using machine learning in frequency domain," in *2015 IEEE 4th Global Conference on Consumer Electronics (GCCE)*. IEEE, 2015, pp. 185–186.

[46] N. Vashistha, H. Lu, Q. Shi, M. T. Rahman, H. Shen, D. L. Woodard, N. Asadizanjani, and M. Tehranipoor, "Trojan scanner: Detecting hardware trojans with rapid sem imaging combined with image processing and machine learning," in *ISTFA 2018: Proceedings from the 44th International Symposium for Testing and Failure Analysis*. ASM International, 2018, p. 256.

[47] K. Hasegawa, M. Oya, M. Yanagisawa, and N. Togawa, "Hardware trojans classification for gate-level netlists based on machine learning," in *2016 IEEE 22nd International Symposium on On-Line Testing and Robust System Design (IOLTS)*. IEEE, 2016, pp. 203–206.

[48] Z. István, G. Alonso, and A. Singla, "Providing multi-tenant services with fpgas: Case study on a key-value store," in *2018 28th International Conference on Field Programmable Logic and Applications (FPL)*. IEEE, 2018, pp. 119–1195.

[49] R. Elnaggar, R. Karri, and K. Chakrabarty, "Multi-tenant fpga-based reconfigurable systems: Attacks and defenses," in *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2019, pp. 7–12.

[50] Inside the microsoft fpga-based configurable cloud. [Online]. Available: https://azure.microsoft.com/en-us/resources/videos/build-2017-inside-the-microsoft-fpga-based-configurable-cloud/

[51] Enable faster fpga accelerator development and deployment in the cloud. [Online]. Available: https://aws.amazon.com/ec2/instance-types/f1/

[52] Intel launches new datacenter accelerator with its most powerful fpga. [Online]. Available: https://www.top500.org/news/intel-launches-new-datacenter-accelerator-with-its-most-powerful-fpga/

[53] A. Khawaja, J. Landgraf, R. Prakash, M. Wei, E. Schkufza, and C. J. Rossbach, "Sharing, protection, and compatibility for reconfigurable fabric with amorphos," in *13th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 18)*, 2018, pp. 107–127.

[54] S. Tian and J. Szefer, "Temporal thermal covert channels in cloud fpgas," in *Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*. ACM, 2019, pp. 298–303.

[55] Y. Luo and X. Xu "HILL: A Hardware Isolation Framework against Information Leakage on Multi-Tenant FPGA Long-Wires," in *2019 International Conference on Field-Programmable Technology (FPT) 2019*. IEEE.

[56] Rajendran J, Sinanoglu O, Karri R. "VLSI testing based security metric for IC camouflaging," in *2013 IEEE International Test Conference (ITC)*. IEEE.

[57] Li M, Shamsi K, Meade T, Zhao Z, Yu B, Jin Y, Pan DZ, "Provably secure camouflaging strategy for IC protection," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 2017 Sep 7.

[58] M. Zhao and G. E. Suh, "Fpga-based remote power side-channel attacks," in *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2018, pp. 229–244.

[59] Xu X, Rührmair U, Holcomb DE, Burleson W, "Security evaluation and enhancement of bistable ring PUF," in *International Workshop on Radio Frequency Identification: Security and Privacy Issues* 2015.

[60] F. Schellenberg, D. R. Gnad, A. Moradi, and M. B. Tahoori, "An inside job: Remote power analysis attacks on fpgas," in *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2018, pp. 1111–1116.

[61] C. Ramesh, S. B. Patil, S. N. Dhanuskodi, G. Provelengios, S. Pillement, D. Holcomb, and R. Tessier, "Fpga side channel attacks without physical access," in *2018 IEEE 26th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, 2018.

[62] I. Giechaskiel, K. B. Rasmussen, and K. Eguro, "Leaky wires: Information leakage and covert communication between fpga long wires," in *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*. ACM, 2018, pp. 15–27.

[63] Xu X, Burleson W, Holcomb DE, "Using statistical models to improve the reliability of delay-based PUFs," in *IEEE Computer Society Annual Symposium on VLSI (ISVLSI)* 2016.

[64] J. Krautter, D. R. Gnad, and M. B. Tahoori, "Fpgahammer: remote voltage fault attacks on shared fpgas, suitable for dfa on aes," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 44–68, 2018.

[65] D. R. Gnad, F. Oboril, and M. B. Tahoori, "Voltage drop-based fault attacks on fpgas using valid bitstreams," in *2017 27th International Conference on Field Programmable Logic and Applications (FPL)*. IEEE, 2017, pp. 1–7.

[66] D. Mahmoud and M. Stojilović, "Timing violation induced faults in multi-tenant fpgas," in *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2019, pp. 1745–1750.