

Article

Survey of Security Issues in Memristor-Based Machine Learning Accelerators for RF Analysis

Will Lillis, Max Cohen Hoffing and Wayne Burleson *

ECE Department, University of Massachusetts Amherst, Amherst, MA 01002, USA; wlillis@umass.edu (W.L.); maxcohenhoffing@outlook.com (M.C.H.)

* Correspondence: burleson@umass.edu

Abstract: We explore security aspects of a new computing paradigm that combines novel memristors and traditional Complimentary Metal Oxide Semiconductor (CMOS) to construct a highly efficient analog and/or digital fabric that is especially well-suited to Machine Learning (ML) inference processors for Radio Frequency (RF) signals. Analog and/or hybrid hardware designed for such application areas follows different constraints from that of traditional CMOS. This paradigm shift allows for enhanced capabilities but also introduces novel attack surfaces. Memristors have different properties than traditional CMOS which can potentially be exploited by attackers. In addition, the mixed signal approximate computing model has different vulnerabilities than traditional digital implementations. However both the memristor and the ML computation can be leveraged to create security mechanisms and countermeasures ranging from lightweight cryptography, identifiers (e.g., Physically Unclonable Functions (PUFs), fingerprints, and watermarks), entropy sources, hardware obfuscation and leakage/attack detection methods. Three different threat models are proposed: (1) Supply Chain, (2) Physical Attacks, and (3) Remote Attacks. For each threat model, potential vulnerabilities and defenses are identified. This survey reviews a variety of recent work from the hardware and ML security literature and proposes open problems for both attack and defense. The survey emphasizes the growing area of RF signal analysis and identification in terms of commercial space, as well as military applications and threat models. We differ from other recent surveys that target ML, in general, neglecting RF applications.



Citation: Lillis, W.; Hoffing, M.C.; Burleson, W. Survey of Security Issues in Memristor-Based Machine Learning Accelerators for RF Analysis. *Chips* **2024**, *3*, 196–215. <https://doi.org/10.3390/chips3020009>

Academic Editors: Gerard Ghibaudo and Francis Balestra

Received: 22 April 2024

Revised: 7 June 2024

Accepted: 8 June 2024

Published: 13 June 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: memristors; eNVM; compute-in-memory; analog-edge processors; RF; machine learning

1. Introduction

With the recent impact of ML in fields such as Robotics, Autonomous Vehicles, Natural Language Processing, and RF analysis among others, practical ML technologies are developing at an astonishing rate. Encouraged by a long list of successful applications in a wide variety of fields, investment in ML technologies (in particular, various Neural Network architectures) is expected to continue its explosive growth for the foreseeable future. As ML technology becomes pervasive and increasingly integrated into our lives, the safety and protection of critical information in ML systems is paramount. Security analysis and development of countermeasures should be performed in a holistic manner since many attacks transcend layers of computation, spanning from algorithms to hardware.

The application of Neural Networks (NNs) to the RF domain (particularly on the edge) exposes several fundamental limitations of computing paradigms designed around traditional CMOS hardware. The data-intensive nature of NN applications is problematic for both power and speed, fundamentally driven by memory-related limitations of the von-Neumann architecture. Deployment to the edge often introduces strict power constraints that conflict with such memory-intensive applications. In addition, the analog nature of raw RF data make them especially difficult to process with traditional CMOS, owing to the necessary conversions to and from the digital domain. Memristors are an

emergent nanoelectric technology that offers numerous benefits over traditional CMOS when deployed in large arrays, including acceleration and efficiency gains in NN computation. Memristor-based accelerators have already demonstrated particular effectiveness in Edge-deployed NN inference roles due to their speed and power efficiency [1]. Recent work indicates that device-level characteristics of memristive technology makes it especially well-suited for RF applications [2].

As integration technology matures, memristors will likely be integrated into hybrid chiplet-based systems that combine multiple technological paradigms with emergent volatile and non-volatile memory technologies that implement analog sensing front-ends. Hybrid designs can include standard digital CMOS (typically programmable Central Processing Units (CPUs), Graphical Processing Units (GPUs), Field Programmable Gate Arrays (FPGAs), and/or Application Specific Integrated Circuits (ASICs)). Trends indicate increasing investment into chiplet architectures to allow more yield-efficient die sizes as well as appropriate technologies, and this will likely be reflected in memristor technology moving forward [3,4]. Memristors, however, raise several new concerns and capabilities for security; including some specific to the analog approach to ML computation, and the non-volatile storage aspects, as well as the overall deployment platforms and application areas [5].

This survey emphasizes the growing area of RF signal analysis and identification for commercial, military and law enforcement spaces, and identifies the relevant threat models and countermeasures in each of those spaces. This contrasts with other recent security surveys of memristors for ML [6] that target ML in general. RF signal analysis and identification differs from other ML applications in several ways:

1. Data rates from an RF front-end can be very high and involve numerous channels.
2. The number of different devices, signals and protocols to be identified and classified can vary considerably.
3. Low-latency computation is required to enable fast real-time response, which can be critical for both application performance (e.g., cognitive radio) and for identification of and defense against active attacks.
4. Application scenarios can involve several different parties producing and analyzing signals with different threat models.
5. Domain-specific challenges currently separate ML applications to RF from more general approaches to ML (Section 4).

We present threat models for memristor-based ML Edge computing for RF Analysis under three different scenarios: Supply Chain Attacks, Physical attacks and Remote attacks. Combinations of the three threat models are also considered. The standard procedure in threat modeling is to identify system assets and system vulnerabilities, and then define security requirements, pinpoint the capabilities and motives of the attacker and then devise potential countermeasures and defenses [7]. While we cover a wide range of vulnerabilities, attacks, and defenses, our survey is not an all-inclusive survey and is intentionally tailored to the RF signal analysis space. Some threats and security capabilities that involve multiple levels of abstraction are investigated (e.g., hardware, software, algorithm and application).

The paper is organized as follows: In Section 2, we provide background on memristors as used in ML computation. In Section 3, we discuss various ML accelerators, focusing on key differences with more traditional computing models. In Section 4, we consider the applications of Edge Computing, in particular for RF Signature Analysis. Section 5 introduces and provides a description of the three threat models: (1) Supply Chain Attacks, (2) Physical Attacks and (3) Remote Attacks. For each threat model, we identify potential vulnerabilities and defenses. Section 6 discusses potential security countermeasures for memristor architectures. Section 7 describes the handling of anomalies and faults in memristor nodes. Section 8 discusses strategies for integrating memristors with more traditional CMOS technology, as well as the associated additional security concerns and defenses caused by such integration. Conclusions are summarized in Section 9.

2. Background on Memristor Technology

Memristors are an emergent device technology, primarily considered as a Non-Volatile memory device. The device was initially proposed by Leon Chua in 1971 as the fourth fundamental circuit element which is “as basic as the three classical circuit elements already in existence, namely, the resistor, inductor, and capacitor” [8]. Memristors are typically constructed with resistive switching materials that exhibit hysteresis. Resistive switching materials currently utilize one of four physical mechanisms to achieve hysteresis: redox reactions, phase transitions, spin-polarized tunneling and ferroelectric polarization. Each mechanism provides a set of trade-offs, with different values for switching speed, switching energy, physical size, data retention and endurance, conductance range, number of distinguishable conductance states, current-voltage linearity, and other relevant properties, especially for memory applications [1].

Due to the increasing strain on sustaining Moore’s law reflected in current CMOS devices and circuitry, the rise in popularity of data-intensive tasks such as ML, and the limits imposed by the “von-Neumann bottleneck”, in certain contexts in memory computing presents itself as an attractive alternative to more traditional schemes [1,9]. Traditionally expensive operations such as vector-matrix multiplication can be completed in parallel, resulting in $\mathcal{O}(1)$ time, by leveraging physical laws; namely Kirchhoff’s Law for summation and Ohm’s Law for multiplication. Recent advances are now demonstrating the capabilities of these devices, including wireless communications [10], fully analog ML accelerators [11] and greatly expanded storage density [12].

Given this set of unique properties and capabilities, memristors are well-suited for ML acceleration roles. In particular, memristive crossbars have been successfully deployed in NN training and inference roles [13]. Figure 1 shows the basic principles of a memristive crossbar array used to perform matrix-vector multiplication in an analog domain.

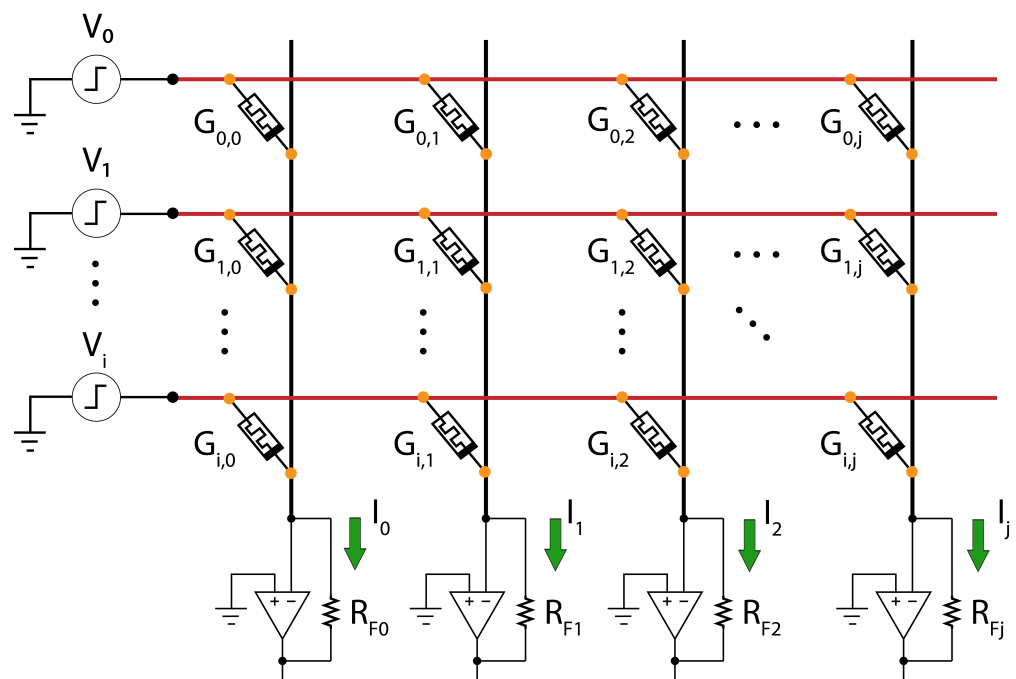


Figure 1. An example of a memristor array. This array shows the basics of a current summation model. The voltage inputs are multiplied against the conductances ($G_{i,j}$) of each device that generates current. The currents are then accumulated at each output.

3. Background on ML Accelerators

Deep Learning has recently made great strides in a wide variety of applications such as object recognition and natural language processing [14]. Along with the increased expressiveness and capabilities of these models, however, comes a rising need for more powerful

yet energy-efficient processors, especially in Internet of Things (IoT) and Edge applications. To accommodate these new computational demands, new hardware (e.g., enterprise GPUs and custom digital ASICs) have been developed to optimize both training and inference of these models. However, these accelerators are still limited by the von-Neumann architecture; hampered by “the constant data shuttling between the information processing and memory units. In addition, the performance mismatch between the processor and memory units leads to considerable latency (commonly referred to as the ‘memory wall’)” [9]. In response to these challenges, increased attention has been given to in situ, or in memory computing: an alternative schema in which memory is used for both storage and computation [15]. In particular, memristive crossbar arrays have been successfully used as in-memory analog ML accelerators [9,11,13].

Approximate Computing

The in situ computing offered by memristive crossbar arrays is a form of approximate computing, in which a shift is made “from conventional precise processing to inexact computation but still satisfying the system requirement on accuracy” [16]. Memristor computation falls into this category due to the inherently inexact nature of analog compared to classical digital computation. While this new paradigm offers performance improvements, it also exposes several new attack surfaces. Yellu et al. [16] introduces four new potential attack scenarios: Building Covert Channel, Error Compensation, Tampering Error Resilience Mechanism, and Error Propagation attacks. Building Covert Channel attacks involve the exploitation of approximate compute modules in order to open passive side channels to enable the leakage of covert information. Error Compensation attacks exploit the imprecise nature inherent to approximate computing to insert stealthy malicious behavior. Tampering Error Resilience Mechanism attacks are inserted into and around the system’s Error Monitor, Precision Cutoff Threshold, and Accuracy Tuner, generally for the purpose of denial-of-service or reverse engineering. Finally, Error Propagation attacks allow the adversary to utilize unsecured approximate-computing modules to forward errors through the rest of the data path. The authors stress the need for new accuracy metrics to be developed in order to differentiate benign from malicious errors. Further research is required to investigate these potential threats and countermeasures with modern and/or future applications in mind.

4. RF Signature Analysis

4.1. Background

As ML (and, in particular, deep learning) becomes ever-more ubiquitous [14], the same is true for the RF analysis domain. In particular, “deep feature learners with an inherent recurrent structure have been shown to perform well” for RF ML tasks [17]. Certain ML architectures have demonstrated strong performance in analyzing different RF signal features. Deep Neural Networks (DNNs) have performed well for fixed value parameters such as rise time, Convolutional Neural Networks have performed well for spatially correlated data such as modulation techniques, and Recurrent Neural Networks have performed well with temporally correlated data such as I/Q signals [17]. LeCun et al. argue that deep learning is a technique that “allows a machine to be fed with large amounts of raw data and to automatically discover the representations needed for detection or classification” [14]. As with other challenging application domains, the rich expressiveness of deep learning models has allowed success where other ML architectures have fallen short. However, the effective application of ML to the RF space is still a non-trivial problem. The inherently noisy nature of RF communications, variation in environmental conditions caused by the ‘train on one day test on another day’ problem [18], lossy conversions between the analog and digital domains, and other issues complicate matters greatly. Extensive dataset and feature engineering (e.g., adding noise, selectively removing parts of the signal, or factoring in “significant modeling insights” [19]) has continued to prove

necessary to achieve acceptable performance in recent work. Various commercializations of this technology target lucrative markets such as spectrum-awareness [20].

4.2. RF Fingerprinting

One relevant application of RF signature analysis is wireless fingerprinting. This technique aims to exploit the hardware variations in analog circuitry, which are then reflected in the signals transmitted by said hardware. These variations can enable the unique identification of different wireless transmitters, regardless of the actual data content of their transmission. The ability to differentiate between multiple (potentially adversarial) transmitters is a powerful physical layer technique, allowing one to detect imposters even after security protocols farther up the network stack have been compromised. Figure 2 illustrates the basic concept of RF Signal identification. Different approaches have utilized both the transient and steady-state portion of the signal, predefined (e.g., modulation errors) and inferred features (e.g., information extracted as a result of applying an FFT), as well as single or multiple features. Ideally, a wireless fingerprint is robust to variations such as environmental conditions, voltage and power levels, relative orientation, and physical distance between the transmitter and receiver [21]. The use of complex-valued DNNs in combination with various feature “augmentation strategies” has shown promise in developing protocol agnostic and variation-resilient fingerprinting techniques [19,22,23].

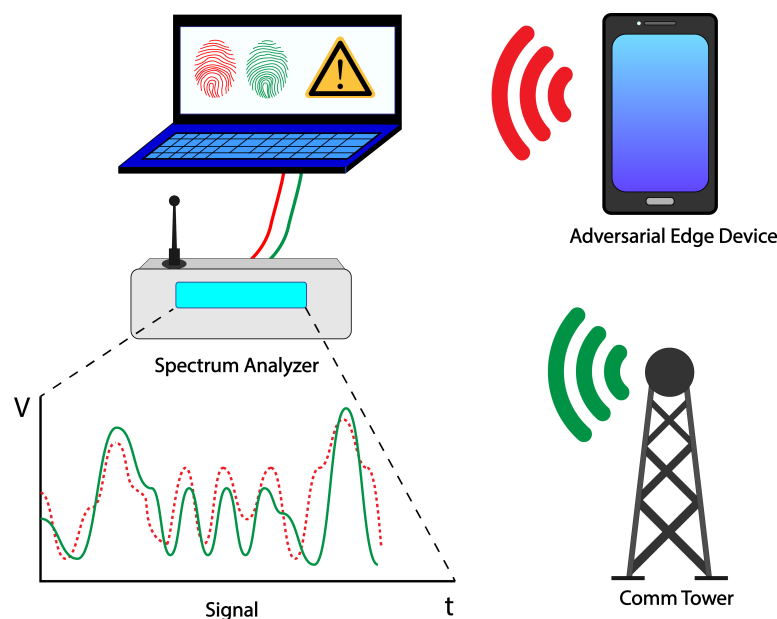


Figure 2. An example of a spectrum analyzer measuring signals, and an ML classifier developed to differentiate between transmitters sending a friendly signal and an adversarial signal based on inherent variation in transmitter properties. The green signal corresponds to a known transmitter, while the red signal corresponds to an adversarial transmitter attempting to impersonate the known transmitter. By utilizing RF fingerprinting techniques, the receiver can potentially differentiate between the two.

Autoencoder pre-training has been shown to mitigate the effectiveness of adversarial examples in the RF space. Convolutional Neural Networks have also shown promise in the application of wireless transmitter fingerprinting. Complex-valued weights have shown advantages over their real-valued counterparts in these tasks since RF data are commonly represented in a complex form. As demonstrated by Wang et al., memristor technology is fully capable of processing complex-valued data and takes advantage of these specialized architectures [10]. Difficulties still exist in the elimination of confounding factors. In addition, the challenges of adversarial learning require a more thorough exploration [19,22].

4.3. Cognitive Radio

In addition to the wireless fingerprinting of one or several wireless entities, it is also worthwhile to take a broader view of the wireless medium a device is operating within. Cognitive radio allows more efficient use of the RF spectrum by dynamically adapting parameters to the current state of a given device's wireless environment. Indeed, such RF networks are "extremely agile, allowing communications to move around the frequency domain optimizing capacity when users are present. These systems require components that can quickly and accurately detect whether or not signals are present in their current channel and what channels are free around them" [20]. More efficient use of the shared wireless medium will become increasingly important as the number of wireless devices continues to increase [24].

4.4. Real World Applications

Memristive crossbars show great promise in non-ML RF applications. By connecting memristive crossbars directly with analog wireless media, Wang et al. were able to "store, transfer and modulate time-varying analog signals and parallelly process data without digital-to-analog converters (DACs) and analog-to-digital converters (ADCs)" [10]. This integration yielded an increase in power efficiency of over 100-fold. Zen et al. created a CMOS-memristor hybrid baseband processor, "implementing a full-fledged communication system" that outperforms its CMOS-only counterparts in speed and energy efficiency by factors of 10^3 and 10^6 , respectively [25]. The recent progress in implementing various ML architectures via memristive crossbar arrays [9,11,13] shows promise for more tightly integrated RF Machine Learning systems in the future.

Given the natural application of memristor technology to the RF space, it is reasonable to develop threat models within the RF domain and its applications. With this in mind, we develop three threat models in the section below.

5. Threat Models

The development of an appropriate threat model is highly context-dependent. In the case of memristor-based analog ML accelerators, the context is derived from the edge device manufacturing process, deployment location, and how it interacts with the larger system into which it is integrated. Threat models for memristors should also consider how an adversary might be able to access the critical assets, namely the information stored in memristors. Access to memristors is split into three categories: White Box (WB), Grey Box (GB) and Black Box (BB). WB access assumes the adversary can provide arbitrary inputs to the model, has prior knowledge about the NN model such as its architecture, and can read the resulting output. Furthermore, an adversary with WB access can read the conductances of individual memristors within the array. BB access assumes the adversary can read input/output pairs, but the adversary has no additional system access. GB access lies between the two, characterizing the adversary as only having partial access to the system. When developing a threat model, it is important to consider who the potential adversaries are and their objectives. For example, the adversary could be a nation-state or commercial competitor. Such an adversary's end goal could be stealing vital technology or inhibiting and sabotaging competition in the market. It then begs the question: What exactly might an adversary try to steal from a memristor-enabled device?

A report by Cottier predicts that the already significant cost associated with training large ML systems will only continue to rise. By 2030, training an AI model could cost as much as \$500 million [26] and requires access to enormous data sets. Reverse engineering others' architectures, hyperparameters, weights and other details offers a considerably more cost-effective path compared to the alternative of developing and training a model independently. Thus, there is a massive economic incentive to steal a trained NN, including the model's weights. In the case of RF technology, the 6G network currently being developed will likely utilize ML technology for use cases such as network mobility, intelligent routing, energy efficiency, etc. Du et al. explain additional avenues in which ML might be

implemented in 6G communication technology [27]. Considering the non-negligible role NNs will play in this space, there are significant economic incentives to provide low-cost solutions for 6G network infrastructure. As a result, many competitors in the commercial sector have a proper economic motivation to steal NN models and their respective weights.

In a military environment, there is not just an economic incentive to protect Intellectual Property (IP) from theft, but also a requirement to protect national security and ensure successful missions. In wireless communication, memristors could aid RF signal processing, communication, and signal classification tasks. In addition to the use cases mentioned above in Section 4, memristors can be used to accelerate RF ML signal classification, such as identification of friend or foe or target tracking. Another major concern in wireless technology for the military community is the impersonation of an ally through replay attacks. In recent history, it was reported that Iran was able to capture a US drone by jamming its sensors and then spoofing its GPS coordinates [28]. Unfortunately, memristors have the potential to increase the attack surface of chip technology as compared to its traditional CMOS counterparts; especially since they will be embedded and integrated with CMOS technology. Jamming, spoofing and replay attacks are just a few attack scenarios discussed under the Remote Attack threat model.

Below, we identify and explore three vital threat models: (1) Supply Chain Attacks, (2) Physical Attacks, and (3) Remote attacks. For each model, we discuss potential methods an adversary could utilize to attack memristor-based technology as well as potential defenses. Following this discussion of individual threat models, we also explore how vulnerabilities may emerge from a combination of threat models.

5.1. Supply Chain Attacks

Security threats in this category can have WB Access, GB Access, or BB Access depending on the location of the attack within a supply chain. Supply chain security is an increasing concern due to the global nature of semiconductor and electronics manufacturing. Figure 3 shows the global nature of the semiconductor supply chain and its vulnerabilities at each stage. Department of Defense (DoD) requirements for chips include a wide variety of technologies that range from legacy microelectronics (node feature sizes greater than 100 nm) to the current state of the art below 5 nm. The DoD does not have standardized technological requirements, creating a large attack surface area via its numerous supply chains. Dependence on a highly concentrated supply chain as opposed to a diverse set of suppliers leaves unnecessary vulnerabilities [29]. The foremost concern in a supply chain threat model is the insertion of counterfeit components, either in high volume for economic benefit or more selectively inserted for strategic benefit.

The presence of counterfeit chips or boards in the supply chain can result in loss of revenue, intellectual property, brand recognition, and reliability in a critical system's functional capability [30]. Various mitigation techniques have been developed for CMOS systems, all of which offer different security properties, deployability and the necessary level of effort to integrate into an existing integrated circuit. See [30] for an in-depth examination.

Counterfeits also include the added threat of insertion of hardware Trojans and other malware. "Hardware Trojans are malicious modifications of a circuit... that aim at manipulating its behavior in an undesired manner". Possible results include denial-of-service, change in functionality, or the opening of side channels to leak sensitive information. Hardware Trojans are by design difficult to detect via traditional methods including testing of the circuit, optical inspection, or side-channel analysis [31]. An adversary can insert a hardware Trojan at any point in the integrated circuit supply chain, including during the in-house design process, 3rd Party IP vendors, Computer Aided Design (CAD) tools, fabrication, testing, and/or distribution phases. Insertion at the fabrication stage is the most common concern [32]. In the case of edge analog accelerators, such malicious alterations to the circuit could lead to issues of classification error, leakage of sensitive data, or device failure. Such issues could appear consistently, intermittently, probabilistically, or only when certain conditions are met in the case of a parametric hardware Trojan. A

parametric hardware Trojan is “carefully inserted to modify the electrical characteristics of predetermined transistors in a circuit by altering parameters such as doping concentration and dopant area” [31]. Presently, the consideration of hardware Trojans in circuits with memristors has been very limited so far. Initial work has highlighted the increased difficulty in distinguishing between sneak path current and effects due to a stealthy hardware Trojan in OT1R designs [33]. Due to the lack of research in this area presently, hardware Trojans currently pose an unaddressed threat in memristive applications.

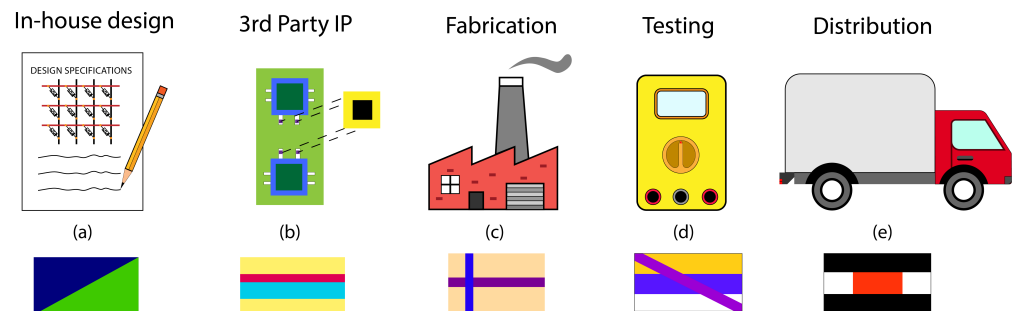


Figure 3. Above is a list of areas in which an adversary could attack during the supply chain. In each step of the supply chain, an attack could occur in a different country or region with different laws and treaties, which increases the area of the attack surface. Note that these are just examples and do not refer to actual countries. (a) Given access to the original design specifications, the adversary could insert side channels to facilitate leakage of valuable information (Country A). (b) Third party products may come from untrustworthy sources, and could contain Trojans or other malicious hardware (Country B). (c) An adversary could produce counterfeit products or compromise the integrity of the chips (Country C). (d) A testing team could reverse engineer the NN model details (e.g., architecture, weights, etc.) by recording the inputs and outputs of the memristor array (Country D). (e) Distributor could illegally retain the chip and/or replace it with a counterfeit (Country E).

Weight programming introduces an instance of vulnerability made possible by the nature of accessing memristors through its peripherals. Memristor crossbar peripherals, such as the programming/readout ADCs are susceptible to probing. As discussed by Huang et al., the memristor “array is typically equipped with high-resolution ADC for the write/verify scheme to minimize the variations of the cell conductance. Individual cell conductance will be read out by such ADC and compared with the reference when initially loading the DNN models on-chip” [34]. Therefore, if an adversary can observe weight programming (e.g., during a manufacturer’s testing phase), then they can easily extract memristor weights.

5.2. Physical Attacks

Gaining physical access to a device opens up numerous attack surfaces to attackers. This translates to GB access to the relevant model in the best case and WB access in the worst case. Assuming a nation-state adversary that possesses technological capabilities (such as an advanced electronics and optical measurement lab), then the potential for a large array of attacks that can be carried out is introduced. Probing allows sensitive data, such as network architecture, proprietary weights, or other pieces of sensitive intellectual property to be stolen.

Huang et al. describe a micro-probing attack in which an adversary with WB access can extract individual weights from a crossbar. Even if the individual cells can not be probed directly, the threat of having their values read out via the crossbar’s peripheral circuitry remains [34]. Such an attack would enable an adversary to steal an entire NN model.

Even under a weaker threat model where entire memristive crossbars are treated as black boxes (and the probing of individual memristors is not viable), Wang et al. demonstrate the danger posed by power and time side-channel leakage. Treating the tiles as black boxes and without prior knowledge of the network architecture, details such as “stored

layer type, layer sequence, output channel/feature size and convolution kernel size” were able to be reverse engineered [35].

A fault injection attack could be carried out by altering the network’s weights, hence reducing inference accuracy. Forensic analysis and reverse engineering could grant an adversary valuable information in carrying out other attacks, such as crafting adversarial inputs to an RF classifier [36]. A captured device also presents numerous opportunities for reverse engineering, from specific circuits and architectures to recovering stored weights and data [37].

Model extraction attacks describe the process of an adversary reverse engineering an ML model by collecting many input/output pairs and using those pairs to train a new model. Such attacks pose a threat to all inference engines since such attacks simply require black-box access to gather sufficient input-output pairs. This threat has been specifically addressed for memristor-based inference accelerators by Huang et al., who explore the threat as well as memristor-specific countermeasures [34]. This attack is not limited to captured devices, as it is also viable in remote attacks if the adversary can obtain enough input/output pairs.

5.3. Remote Attacks

Remote attacks encompass any adversarial action at a distance where physical access to the device is not available. Previous works [17,36] have demonstrated the danger of adversarial inputs to ML models in RF applications. The threat of adversarial inputs is not only limited to general misclassification but also includes the erroneous classification of a specific target class. Jamming attacks enable an adversary to achieve denial-of-service for receiving and/or transmitting data. All wireless media is vulnerable to jamming due to the shared nature of the wireless medium. Identification of jamming attacks can aid in avoiding or subverting them (e.g., via frequency hopping, reconfigurable antennas, or application of cognitive radio). ML techniques have shown promise in accomplishing said identification [38].

As with any application that utilizes RF communications, a further concern is that of impersonation and replay attacks. Countermeasures must be taken in order to prevent an adversary from posing as a trusted entity; whether that be via novel techniques at the memristor/crossbar level or by utilizing standard cryptographic techniques higher up the stack (e.g., Message Authentication Code schemes) [7].

5.4. Hybrid Attacks

In addition to threats from the above models, a hybrid attack utilizing multiple categories should be considered. A hybrid attack is more likely to result in successful penetration or denial of service since it combines the attack surfaces available to multiple threat models. For example: an adversary could obtain a device, reverse engineer intimate model information stored on the device (such as weight matrix values) and use that information to launch devastating attacks remotely [36]. Another example of a hybrid threat is where an adversary inserts a parametric hardware Trojan into the weight matrix (e.g., via a compromised supply chain) and then uses the perturbations to increase the effects of other attacks (e.g., jamming) after the device’s deployment. Other scenarios can be envisioned where attacks in one threat model can be used to either learn information or modify functionality to facilitate later attacks when deployed in the field.

6. Countermeasures

The system of RF/ML/Memristor computation has several interesting security features in addition to the vulnerabilities discussed above. These include mechanisms for unique identifiers, key generation, true random number generation, lightweight cryptography, and obfuscation of hardware, software, and ML models. These can all be used as countermeasures to the attacks described above. Below, we describe existing work and open problems in several of these areas.

6.1. Physically Unclonable Functions (PUFs)

A PUF is a relatively new hardware-based security primitive that “exploit[s] the ... manufacturing variations that occur in almost all physical systems on small length scales”. By design, a PUF is “unclonable, and constitutes an individual fingerprint of each system” [39]. PUFs can serve a variety of purposes, including identification and authentication, key storage, key exchange, digital rights management, tamper detection, and memory encryption [39]. Existing PUFs can be modified slightly to provide these features to memristive circuits. In particular, PUFs serve as a natural countermeasure to counterfeiting attacks.

Memristors in various configurations have also been used to construct hybrid PUFs, promising better performance than their CMOS-only counterparts. Koeberl et al. utilize a “weak write mechanism”, in which insufficient access time is given to a memristor write operation, leaving the device’s conductance in an undefined state determined by device variation [40]. Gao et al. utilize the variation of individual memristors within a large crossbar array in combination with ring oscillators to “translate a memristance value into a frequency through a [current mirror-controlled ring oscillator]” in order to generate random bits [41]. Zhang et al. present a silver sputtering technique in which a memristive crossbar serves as a weak PUF, producing a single challenge-response pair [42]. Several other weak PUF designs utilize differential pairs of memristors [43–45], which compare the conductances of pairs of memristors. Unfortunately, this particular approach requires two (2) crossbars for each bit of key. Moreover, these PUFs do not prevent an adversary from learning the fingerprint through probing from a memristor array, and therefore, only provide a weak security property.

Govindaraj et al. describe an Arbiter PUF design where memristor cells are “connected in daisy chain fashion selectively between two symmetric paths” [46]. Chatterjee et al. attempt to harden another memristor-based Arbiter PUF design [47] against cryptanalysis [48]. Uddin et al. leverage experimental measurements of hafnium-oxide memristors, exclusive OR (XOR)ing, and a column shuffling technique to construct an improved memristor-based Arbiter PUF, designated as XbarPUF [49]. This technology is still immature and has limited availability for testing outside of simulation. Additionally, memristor-based PUFs were found to be highly susceptible to modeling attacks in many cases [5]. Ibrahim et al. fabricated a memristor-based PUF, allowing for full testing outside of simulation [50]. More recently, Al-Timimi et al. simulated a novel memristor PUF design in simulation that attempted to minimize the negative effects of the sneak path currents [51].

Providing security measures for memristor technology that can detect and combat probing or tampering remains an open problem.

6.2. Lightweight Cryptography

Traditional cryptographic algorithms and protocols are often ill-suited for resource-constrained devices. The recent explosion in IoT devices has led to the development of new lightweight cryptographic protocols for encryption, authentication, integrity checking, and key management. As a result, there are numerous lightweight protocols tailored to various use cases, such as PRINCE, PRESENT, and lightweight variants of traditional algorithms such as Advanced Encryption Standard (AES) and Data Encryption Standard (DES). For an exhaustive listing, see Rana et al. [52].

The usage of memristors to provide efficient crypto-computation in resource-constrained settings has been considered, as in Xue et al.’s proposal to integrate area-efficient in-situ blockchain technology integrated with a RISC-V processor [53]. There are also lighter-weight variants of Public Key Systems, such as Elliptic Curves and some Post-Quantum cryptography algorithms [54].

6.3. Data Flushing

A potential defense mechanism involves the flushing of onboard data once tampering has been detected. This could be employed to prevent the invasive reading (or overwriting)

of proprietary weights in a memristive crossbar array, flushing of secret keys, and flushing of all intermediary values in a matrix-vector multiply operation, etc. Matsuda et al. describe such a flushing mechanism to defend against laser fault injection on an AES cryptographic processor. A bulk built-in current sensor provides the chip with 100% coverage, triggering the flush if a current above a certain threshold is detected in the chip's silicon substrate [55]. While improvement is needed, memristor's CMOS-compatibility [1] shows promise for the integration of similar detection and/or defense mechanisms.

6.4. Model Protection Mechanisms

Deploying Edge NN accelerators in which a model's weights are stored in non-volatile memory poses a heightened risk for various model extraction attacks. There exists a wide range of hardware-agnostic strategies aimed at protecting NN models from unauthorized copying. Lederer et al. provide an extensive exploration of these strategies [56]. Detailing all of these defenses is beyond our scope herein; however, there are a few memristor-centric defenses that we considered, such as [6]. Zou et al.'s survey describe several memristor-specific strategies, including exploiting the memristor's obsolescence effect, model encryption, NN weight transformation, and fingerprint embedding.

The memristor's obsolescence effect describes the gradual degradation of the stored resistance state in an individual device as the result of repeated read voltage pulses. By leveraging this effect, adversaries are prevented from gaining sufficient input/output pairs for a model extraction attack. This strategy requires frequent reprogramming or refreshing of the stored weights, introducing energy overheads, unavoidable delays, and device downtime.

Model encryption schemes store the NN's weights in an encrypted form, decrypting each time they are needed for inference. Example implementations are presented by Zou et al. and Lin et al.'s respective bias [57] and shuffling [58] techniques. Testing of both strategies was constrained to simulation, however, and did not consider the effect of any potential side-channel leakage. In general, this strategy introduces significant latency and energy overheads, both due to the encryption/decryption as well as the frequent reprogramming of the memristor weights. Efforts to optimize this process including limiting encryption to a critical subset of the model's weights have been accomplished [59]. However, an adversary with WB access and granular control over the device could access the weights once they are decrypted for use.

The NN weight transformation strategy aims at protecting weights by altering how they are stored on the physical device. Existing approaches include obfuscating row and/or column connections, either within the same or multiple crossbars.

The fingerprint embedding strategy embeds device-specific information into the programmed NN weights, rendering them useless if transplanted onto another device. Huang et al. explored the use of ADC offsets for this defense strategy [34].

6.5. Design Obfuscation

Design Obfuscation is a technique that shields the hardware details against acquisition by an unauthorized party. This can be used to protect IP, but also to defend against reverse engineering and higher-level attacks. In addition to demonstrating the danger of power and time side channel leakage, Wang et al. also propose several obfuscation-based countermeasures to side-channel attacks. These countermeasures include the scrambling of crossbar tile starting times, dummy inputs to peripheral circuitry components (such as ADCs) and power equalization schemes [35]. Huang et al. [34] present an input shuffling scheme that defends against an input/output pair attack described in Section 5.2.

Logic Locking

Logic Locking is a defense mechanism developed in response to threats made prominent by the globalization of the integrated circuit supply chain. The new era of fabless semiconductor companies requires interaction with multiple third-party (potentially un-

trusted) entities. The threats posed by such a distributed manufacturing process include intellectual property piracy, reverse engineering, and hardware Trojans. Logic locking attempts to add additional security to integrated circuit designs by inserting additional logic into a circuit and locking the original design with a secret key [60]. Jiang et al. propose a logic locking/unlocking scheme with provable key destruction solely with a memristor crossbar array. Such an approach “enables the same crossbar to be used for both security and computing/memory applications, saving chip space while increasing power efficiency” [43].

6.6. Watermarking and Stenography

Watermarking of computation, weights and data provides another security capability. Building on the classic Confidentiality-Authenticity-Integrity (CIA) aspects of security, watermarking hides an identifier into a set of data such that it can not be detected, removed or altered. In this case, output data from a protected ML system might be used illegally by an adversary; however, the provenance of the data can be detected either at run-time or in a forensic analysis through the use of a watermark inserted during the ML inference computation [61]. A watermark should also be robust to modifications of the data such as cropping, compression, noise and other manipulations [61]. Figure 4 shows the basic idea behind adding a watermark to detect the origin of a particular model output.

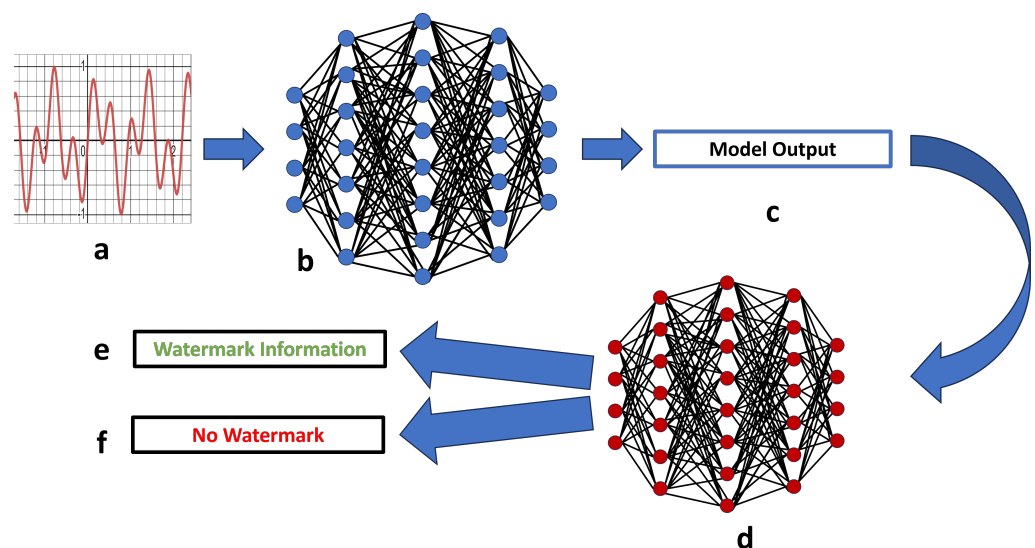


Figure 4. A flow chart depicting the potential use of watermarking to detect imposter/compromised classifiers. An RF signal (a) is received by an inference accelerator (b), which then performs a computation on the signal and sends its output. The output (c) is either stored or immediately forwarded to the second model. The output is received by a watermark extractor (d), which either detects watermarked information in the output (e) or detects the lack of a watermark (f) indicating a compromised device.

The RF ML identification problem clearly can benefit from watermarking to protect model IP, original data, and proprietary algorithms (as well as for forensics). Furthermore, it should be possible to efficiently implement current watermarking algorithms in memristor technology by modifying the stored weights. Co-locating the watermarking with the computation also reduces vulnerabilities. Watermarking can be implemented on the weights or the input/output data. Challenges include implementing the watermarking computation in a fully analog (multi-level) memristor crossbar, as opposed to using only binary nodes [62]. The inherently imprecise nature of memristors will add additional challenges, such as distinguishing a legitimate watermark (or lack thereof) from benign noise or modification by a sophisticated adversary.

Watermarking has been widely implemented with image-based media. Although there are a few image watermarking methods applied to memristors as presented in [63], there has not been extensive research performed on watermarking memristor weights, in general, especially those used in an RF-based context.

The ability to watermark a model's output data could be leveraged to prevent impersonation, replay, or other similar attacks. Chang and Su present an NN-based watermarking scheme in which an input image is fed into a neural network and a watermarked copy of the image is then output [64]. Such a scheme could conceivably be integrated into an existing system by feeding the output neurons of the main network into the inputs of the watermarking network. Huang et al. present a similar scheme in which the watermark could be recovered from a watermarked image without the original image or watermark information known a priori [65].

6.7. Anomaly Detection and Fault Tolerance

Anomalies, whether the result of an adversary or benign, can be detected in a variety of sensors including temperature, voltage, current [55], and signal integrity. In addition to security concerns, the information gathered by such sensors can be used to improve device performance. Zhao et al. demonstrate this by deploying a monitoring infrastructure of sensors to effectively implement dynamic frequency scaling on multicore processors [66].

Due to an immature fabrication process, as well as various device non-idealities, the transfer of an NN model onto a memristive crossbar is not a straightforward task. However, NNs (and, in particular, DNNs) are generally known to be quite fault-tolerant. This useful property is also observed when such architectures are deployed on memristive accelerators, despite various device-related defects and variation. This tolerance is limited, however, as there are typically a non-trivial number of critical faults that can lead to inference errors, including misclassification [67]. Faults due to device non-idealities, manufacturing variations, and failures are of particular interest. Examination of fully hardware memristive NNs shows that small nonlinearity affects accuracy minimally, but large non-linearities can considerably degrade the model's performance [11]. The same model also demonstrated a high tolerance for variance in the hardware-based ReLU's gain.

Sun and Yu examine the effects of memristor non-idealities on both training and inference via simulation, assessing the non-linearity and asymmetry of conductance tuning, manufacturing variations, endurance, and retention. Following this examination, the following conclusions were reached:

- Training accuracy is more sensitive to the asymmetry of conductance tuning than the nonlinearity.
- Conductance range variation does not degrade the training accuracy, instead, a small variation can even reduce the accuracy loss introduced by asymmetry.
- Device-to-device variation can also remedy the accuracy loss due to asymmetry, while cycle-to-cycle variation leads to significant accuracy degradation.
- Different drifting modes affect the inference accuracy differently. The best case is where the conductance is drifting up/down randomly, rather than systematically in one direction [68].

Several ML-based techniques have been developed to identify and prune critical faults from the network. The removal of said critical faults reduces the number of redundant columns necessary for effective fault tolerance by up to 99% [67,69].

Online fault detection is of particular interest for the crossbar arrays, as both adversarial action and material aging effects can lead to a loss of the programmed value for weights in the network. Several methods for online fault detection have been developed to solve this problem, including Chen and Chakrabarty's method that utilizes a watermarking-based backdoor schema [62]. In this schema, the neural network can be checked for faults by periodically attaching watermarks to otherwise normal inputs. Due to the model being over-fitted to the watermarks beforehand, faults have a high likelihood of causing inference errors on said inputs. Unfortunately, present NN watermarking has largely been

conducted using image datasets [61]. Research in the area has not explored applications in the RF domain. More general and efficient online fault detection techniques are in need of development. Anomalies can also be detected through specialized sensor chiplets or package-level sensors in chiplet architectures.

7. Package Security Using Chiplets

While memristors offer tremendous benefits in certain applications, they do not offer the same general-purpose capabilities of typical CMOS systems. In order to realize the benefits of a memristive accelerator in a non-laboratory environment, this technology should be integrated with other more traditional computing paradigms. Current research commonly utilizes custom-built peripheral circuitry to accomplish this task [11,13]. For a more tightly integrated system in a package, a chiplet architecture is a strong candidate instead of a monolithic system. A sample architecture for RF applications is shown in Figure 5 in which a CMOS SoC (System on Chip) handles general computing and a memristor array is directly connected to an RF front end for efficient processing of incoming signals.

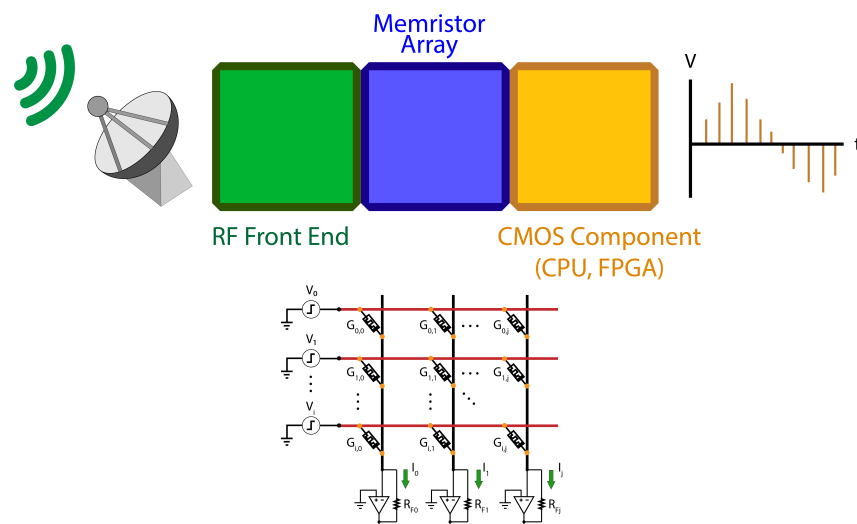


Figure 5. A sample SoC (System on Chip) architecture, integrating RF receivers/transmitters, one or more memristive crossbars for analog signal processing, and a CMOS System to handle general computing. This system can be partitioned into a chiplet architecture, and then later integrated on a common silicon interposer that provides high-speed interconnects between the chiplets as well external interfaces, power delivery and potentially active circuits.

We now discuss vulnerabilities and countermeasures that are specific to chiplet-based systems, referring to the threat models developed in Section 5.

7.1. Chiplet-Based Vulnerabilities

With the benefits of chiplet designs [3] comes “a new frontier of security challenges” [70] that must be addressed, both general and specific to memristors. Concerns include the threat of die swapping (Figure 6), tampering, and the increased ease of invasive probing of the die interconnects. Chiplet security issues fall under both the Supply Chain and Physical Attack threat models.

Counterfeits are a steadily rising threat in the global supply chain. For example, according to the European Union Intellectual Property Office (EUIPO) Crime Tech Report [71], nearly one out of five smartphones sold are counterfeit. Moreover, the loss in global sales due to counterfeit devices averaged to approximately 13%. EUIPO also observed that some regions, such as Latin America, incurred almost 20% in sales losses. In a chiplet model, many of the components will likely be commercial off-the-shelf, and sourcing these parts can be risky without an efficient method to validate authenticity.

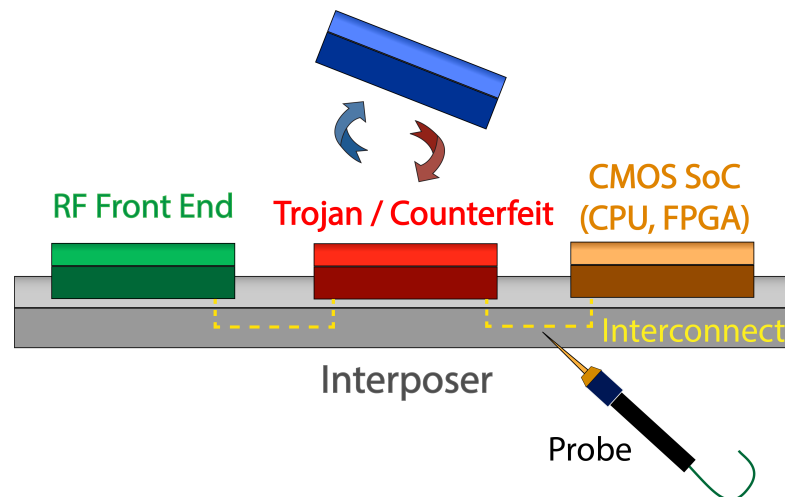


Figure 6. The figure above shows two potential attacks against a chiplet type architecture: Die-swapping, and probing of the interconnections between chiplets.

Hardware Trojans pose a greater risk for chiplets (as opposed to monolithic systems) due to the increased affordability of reverse engineering and/or re-manufacturing of smaller chiplets. This risk is enhanced when considering the increased vulnerability of trailing technology nodes that may be used to save costs in non-critical paths [72]. Hardware Trojans from untrustworthy chiplets may also employ side-channel analysis attacks as demonstrated in [73] in order to obtain valuable information. Chiplets are especially susceptible to this type of attack because the Power Delivery Network (PDN) is shared amongst all of the chiplets [74].

Chiplet architectures are also vulnerable to direct or EM probing since the inter-die wires are larger and more accessible than on-die wires.

7.2. Chiplet Security Countermeasures

One methodology that validates chiplets is proposed by Vashistha et al. [75]. The methodology suggests using a scanning electronic microscope to physically measure, capture, and catalog internal component images and layouts into a database. Although their methodology claims great effectiveness in dealing with tampering, Trojans, and counterfeits, scanning electronic microscope methods are known to be very cost- and time-intensive (sometimes requiring several hours to inspect a single component). Furthermore, decapsulation of the component is often necessary and can be thwarted with tamper-resistance techniques. These downsides will likely prove prohibitive for inspections of volumes of even moderate size.

Other points of adversarial access include probing, decapsulation, delidding, and polishing. Mosavirik et al. demonstrated that nearly any sort of probing, polishing, or merely even scratching the surface of a chiplet will alter the impedance in the PDN [76]. Mosavirik et al. take advantage of this phenomenon by developing a method of authentication that can detect probing, tampering, and hardware Trojans.

Another countermeasure that presents a novel method to authenticate chiplet products is presented by Deric and Holcomb. They propose developing a PUF through delay measurement across the interconnects between chiplets [77]. This authentication strategy could be used as an effective PUF and is more time-efficient compared to [75] above. While this approach is promising, tampering and probing are not examined in detail therein; thus additional exploration to verify PUF robustness is called for.

An industry perspective on chiplet security is presented in [78]. Many of the security challenges and techniques from SoC design apply to Chiplet-based design. In particular, the assembly of and communication between multiple untrusted IP cores, and the protection of the IP in those cores, are similar in SoC and Chiplet-based design. Differences include the unintended use models of chiplets and different business models. Solutions include

the “activation” of chiplets through certificates and the use of blockchain ledgers and other cryptographic techniques [79].

Circuit and software obfuscation can be combined with techniques to protect IP and prevent reverse engineering. Finally, standards are being developed which include security aspects such as Universal Chiplet Interconnect Express (UCIe) [80], the Open Domain-Specific Architecture (ODSA) [81], and Compute Express Link (CXL) [82], as well as network-on-chip approaches.

8. Open Problems

There are numerous open problems in the combination of RF/ML/memristor security. They range from RF datasets and threat models to ML vulnerabilities, to the memristor computational fabric and its packaging into an overall system. Novel attacks and defenses can be envisioned at many levels and across technologies. In many cases, existing work can be leveraged and modified to fit the application and computational assumptions.

A few open problems that have been mentioned earlier include:

- Potential hardware Trojans in memristor technology as well as methods for detection and prevention. As noted in Section 5.1, the threat of hardware Trojans in the context of memristive and memristive-CMOS hybrid technologies is largely unexplored.
- The role of approximate computing in terms of both vulnerabilities and defense [16]. The inexact nature of analog computation allows for a new class of attacks that requires investigation.
- Additional research into efficient in-situ fault detection and recovery for memristor arrays [62].
- More advanced memristor PUF designs, with a particular focus on resilience to modeling attacks [5,83].
- Development of watermarking and design obfuscation techniques that can integrate CMOS with memristor crossbar technology.
- Producing reliable and effective PUFs (see Section 6.1) and TRNGs (see Section 6.2) that utilize memristor architectures.
- A deeper investigation of security vulnerabilities relevant to memristor chiplet architectures (see Section 7).
- Developing more precise threat models for each of the three categories (1) Supply Chain, (2) Physical Attacks, (3) Remote Attacks as the memristor manufacturing process matures.
- Developing on-chip sensors and front-end processing technology specific to memristor crossbar arrays designed for a zero-trust environment.
- Thorough physical testing of memristor-based concepts (e.g., PUFs, fault tolerance schemes, etc.) that have previously been verified only by simulation [5].

9. Conclusions

Memristors have the potential to accelerate RF signal processing in both energy efficiency and speed, especially in Edge scenarios. With these benefits comes the potential for new vulnerabilities, many originating from the device’s ability to store information in a non-volatile manner. With the exorbitant costs required to develop ML models, competitors have every incentive to steal the model architecture and weights. We presented a broad survey on security for memristor architecture with respect to the RF domain. We organized potential adversarial action into three threat models: Supply Chain Attacks, Physical Attacks, and Remote attacks. Next, we described potential defenses and countermeasures for attacks under each of the threat models. We also provided a brief review of security for chiplet architectures since this is a likely format in which memristor technology will be integrated and deployed. Finally, we defined several open problems and potential research directions that should serve to mitigate vulnerabilities and foster robust implementations of memristor technology.

Author Contributions: Conceptualization, W.B.; methodology, W.B.; software, W.L.; validation, W.B.; formal analysis, W.L.; investigation, W.L.; resources, W.B.; data curation, W.L.; writing—original draft preparation, W.L.; writing—review and editing, W.L., M.C.H. and W.B.; visualization, M.C.H.; supervision, W.B.; project administration, W.B.; funding acquisition, W.B. All authors have read and agreed to the published version of the manuscript.

Funding: Research was sponsored in part by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-23-2-0014.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Acknowledgments: Thanks to our fellow researchers Qiangfei Xia, Ali Abdel-Maksoud, and Yi Huang for their technical support.

Conflicts of Interest: The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

1. Wang, Z.; Wu, H.; Burr, G.W.; Hwang, C.S.; Wang, K.L.; Xia, Q.; Yang, J.J. Resistive switching materials for information processing. *Nat. Rev. Mater.* **2020**, *5*, 173–195. [CrossRef]
2. Teja Nibhanupudi, S.S.; Roy, A.; Veksler, D.; Coupin, M.; Matthews, K.C.; Disiena, M.; Ansh; Singh, J.V.; Gearba-Dolcan, I.R.; Warner, J.; et al. Ultra-fast switching memristors based on two-dimensional materials. *Nat. Commun.* **2024**, *15*, 2334. [CrossRef] [PubMed]
3. Sperling, E.; Heyman, K. The March toward Chiplets. *Semiconductor Engineering*. 2022. Available online: <https://semiengineering.com/the-march-toward-chiplets/> (accessed on 28 September 2023).
4. Clark, D. U.S. Focuses on Invigorating ‘Chiplets’ to Stay Cutting-Edge in Tech. *The New York Times*. 2023. Available online: <https://www.nytimes.com/2023/05/11/technology/us-chiplets-tech.html> (accessed on 28 September 2023).
5. Zeitouni, S.; Stapf, E.; Fereidooni, H.; Sadeghi, A.R. On the Security of Strong Memristor-based Physically Unclonable Functions. In Proceedings of the 2020 57th ACM/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 20–24 July 2020; pp. 1–6. [CrossRef]
6. Zou, M.; Du, N.; Kvatinsky, S. Review of security techniques for memristor computing systems. *Front. Electron. Mater.* **2022**, *2*, 1010613. [CrossRef]
7. Anderson, R. *Security Engineering: A Guide to Building Dependable Distributed Systems*; Wiley: Hoboken, NJ, USA, 2020.
8. Chua, L. Memristor-The missing circuit element. *IEEE Trans. Circuit Theory* **1971**, *18*, 507–519. [CrossRef]
9. Xia, Q.; Yang, J.J. Memristive crossbar arrays for brain-inspired computing. *Nat. Mater.* **2019**, *18*, 309–323. [CrossRef] [PubMed]
10. Wang, C.; Ruan, G.J.; Yang, Z.Z.; Yangdong, X.J.; Li, Y.; Wu, L.; Ge, Y.; Zhao, Y.; Pan, C.; Wei, W.; et al. Parallel in-memory wireless computing. *Nat. Electron.* **2023**, *6*, 381–389. [CrossRef]
11. Kiani, F.; Yin, J.; Wang, Z.; Yang, J.J.; Xia, Q. A fully hardware-based memristive multilayer neural network. *Sci. Adv.* **2021**, *7*, eabj4801. [CrossRef]
12. Rao, M.; Tang, H.; Wu, J.; Song, W.; Zhang, M.; Yin, W.; Zhuo, Y.; Kiani, F.; Chen, B.; Jiang, X.; et al. Memristor devices denoised to achieve thousands of conductance levels. *Nature* **2023**, *615*, 823–829. [CrossRef] [PubMed]
13. Li, C.; Belkin, D.; Li, Y.; Yan, P.; Hu, M.; Ge, N.; Jiang, H.; Montgomery, E.; Lin, P.; Wang, Z.; et al. Efficient and self-adaptive in-situ learning in multilayer memristor neural networks. *Nat. Commun.* **2018**, *9*, 2385. [CrossRef]
14. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
15. Xiao, T.P.; Feinberg, B.; Bennett, C.H.; Prabhakar, V.; Saxena, P.; Agrawal, V.; Agarwal, S.; Marinella, M.J. On the Accuracy of Analog Neural Network Inference Accelerators. *IEEE Circuits Syst. Mag.* **2022**, *22*, 26–48. [CrossRef]
16. Yellu, P.; Buell, L.; Mark, M.; Kinsy, M.A.; Xu, D.; Yu, Q. Security Threat Analyses and Attack Models for Approximate Computing Systems: From Hardware and Micro-architecture Perspectives. *ACM Trans. Des. Autom. Electron. Syst.* **2021**, *26*, 1–31. [CrossRef]
17. Roy, D.; Mukherjee, T.; Chatterjee, M. Machine Learning in Adversarial RF Environments. *IEEE Commun. Mag.* **2019**, *57*, 82–87. [CrossRef]
18. Mohanti, S.; Soltani, N.; Sankhe, K.; Jaisinghani, D.; Di Felice, M.; Chowdhury, K. AirID: Injecting a Custom RF Fingerprint for Enhanced UAV Identification using Deep Learning 2020. In Proceedings of the GLOBECOM 2020—2020 IEEE Global Communications Conference, Taipei, Taiwan, 7–11 December 2020; pp. 1–6. [CrossRef]
19. Cekic, M.; Gopalakrishnan, S.; Madhow, U. Wireless Fingerprinting via Deep Learning: The Impact of Confounding Factors. *arXiv* **2021**, arXiv:2002.10791. [CrossRef].

20. Inc, D. Spectrum Awareness. Available online: <https://www.deepsig.ai/spectrum-awareness> (accessed on 25 September 2023).
21. Danev, B.; Zanetti, D.; Capkun, S. On physical-layer identification of wireless devices. *ACM Comput. Surv.* **2012**, *45*, 1–29. [CrossRef]
22. Gopalakrishnan, S.; Cekic, M.; Madhow, U. Robust Wireless Fingerprinting via Complex-Valued Neural Networks. *arXiv* **2019**, arXiv:1905.09388. [CrossRef]
23. Kokalj-Filipovic, S.; Miller, R. Adversarial Examples in RF Deep Learning: Detection of the Attack and its Physical Robustness. *arXiv* **2019**, arXiv:1902.06044. [CrossRef]
24. Muchandi, N.; Khanai, R. Cognitive radio spectrum sensing: A survey. In Proceedings of the 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), Chennai, India, 3–5 March 2016; pp. 3233–3237. [CrossRef]
25. Zeng, Q.; Liu, J.; Lan, J.; Gong, Y.; Wang, Z.; Li, Y.; Huang, K. Realizing Ultra-Fast and Energy-Efficient Baseband Processing Using Analogue Resistive Switching Memory. *arXiv* **2022**, arXiv:2205.03561. [CrossRef]
26. Cottier, B. Trends in the Dollar Training Cost of Machine Learning Systems. *Epoch*. 2023. Available online: <https://epochai.org/blog/trends-in-the-dollar-training-cost-of-machine-learning-systems> (accessed on 22 September 2023).
27. Du, J.; Jiang, C.; Wang, J.; Ren, Y.; Debbah, M. Machine Learning for 6G Wireless Networks: Carrying Forward Enhanced Bandwidth, Massive Access, and Ultrareliable/Low-Latency Service. *IEEE Veh. Technol. Mag.* **2020**, *15*, 122–134. [CrossRef]
28. Peterson, S.; Faramarzi, P. Exclusive: Iran Hijacked US Drone, Says Iranian Engineer. *Christian Science Monitor*. Available online: <https://www.csmonitor.com/World/Middle-East/2011/1215/Exclusive-Iran-hijacked-US-drone-says-Iranian-engineer> (accessed on 22 September 2023).
29. Mondschein, J.; Welburn, J.W.; Gonzales, D. Securing the Microelectronics Supply Chain. *RAND Corporation*. 2022. Available online: <https://www.rand.org/pubs/perspectives/PEA1394-1.html> (accessed on 9 August 2023).
30. Arafat, M.T.; Stanley, A.; Sharma, P. Hardware-based anti-counterfeiting techniques for safeguarding supply chain integrity. In Proceedings of the 2017 IEEE International Symposium on Circuits and Systems (ISCAS), Baltimore, MD, USA, 28–31 May 2017; pp. 1–4. [CrossRef]
31. Kumar, R.; Jovanovic, P.; Burleson, W.; Polian, I. Parametric Trojans for Fault-Injection Attacks on Cryptographic Hardware. In Proceedings of the 2014 Workshop on Fault Diagnosis and Tolerance in Cryptography, Busan, Republic of Korea, 23 September 2014; pp. 18–28. [CrossRef]
32. Xue, M.; Gu, C.; Liu, W.; Yu, S.; O'Neill, M. Ten years of hardware Trojans: A survey from the attacker's perspective. *IET Comput. Digit. Tech.* **2020**, *14*, 231–246. [CrossRef]
33. Basu, S.; Kule, M.; Rahaman, H. Detection of Hardware Trojan in Presence of Sneak Path in Memristive Nanocrossbar Circuits. In Proceedings of the 2021 International Symposium on Devices, Circuits and Systems (ISDCS), Higashihiroshima, Japan, 3–5 March 2021; pp. 1–4. [CrossRef]
34. Huang, S.; Peng, X.; Jiang, H.; Luo, Y.; Yu, S. New Security Challenges on Machine Learning Inference Engine: Chip Cloning and Model Reverse Engineering. *arXiv* **2020**, arXiv:2003.09739. [CrossRef]
35. Wang, Z.; Meng, F.H.; Park, Y.; Eshraghian, J.K.; Lu, W.D. Side-channel attack analysis on in-memory computing architectures. *IEEE Trans. Emerg. Top. Comput.* **2023**, *12*, 109–121. [CrossRef]
36. Kokalj-Filipovic, S.; Miller, R.; Morman, J. Targeted Adversarial Examples Against RF Deep Classifiers. In Proceedings of the WiSec'19: 12th ACM Conference on Security and Privacy in Wireless and Mobile Networks, Miami, FL, USA, 15–17 May 2019; pp. 6–11. [CrossRef]
37. Dhavle, A. Reverse Engineering of Integrated Circuits: Tools and Techniques. *arXiv* **2022**, arXiv:2208.08689. [CrossRef]
38. Jacovic, M.; Rey, X.R.; Mainland, G.; Dandekar, K.R. Mitigating RF jamming attacks at the physical layer with machine learning. *IET Commun.* **2023**, *17*, 12–28. [CrossRef]
39. Ruhrmair, U.; Holcomb, D.E. PUFs at a glance. In Proceedings of the 2014 Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, Germany, 24–28 March 2014; pp. 1–6. [CrossRef]
40. Koerber, P.; Kocabas, U.; Sadeghi, A.R. Memristor PUFs: A New Generation of Memory-based Physically Unclonable Functions. In Proceedings of the 2013 Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, France, 18–22 March 2013; pp. 428–431. [CrossRef]
41. Gao, Y.; Ranasinghe, D.C.; Al-Sarawi, S.F.; Kavehei, O.; Abbott, D. mrPUF: A Novel Memristive Device Based Physical Unclonable Function. In Proceedings of the 13th International Conference, ACNS 2015, New York, NY, USA, 2–5 June 2015; Series Title: Lecture Notes in Computer Science; Volume 9092, pp. 595–615. [CrossRef]
42. Zhang, R.; Jiang, H.; Wang, Z.R.; Lin, P.; Zhuo, Y.; Holcomb, D.; Zhang, D.H.; Yang, J.J.; Xia, Q. Nanoscale diffusive memristor crossbars as physical unclonable functions. *Nanoscale* **2018**, *10*, 2721–2726. [CrossRef] [PubMed]
43. Jiang, H.; Li, C.; Zhang, R.; Yan, P.; Lin, P.; Li, Y.; Yang, J.J.; Holcomb, D.; Xia, Q. A provable key destruction scheme based on memristive crossbar arrays. *Nat. Electron.* **2018**, *1*, 548–554. [CrossRef]
44. Pang, Y.; Wu, H.; Gao, B.; Deng, N.; Wu, D.; Liu, R.; Yu, S.; Chen, A.; Qian, H. Optimization of RRAM-Based Physical Unclonable Function With a Novel Differential Read-Out Method. *IEEE Electron Device Lett.* **2017**, *38*, 168–171. [CrossRef]
45. Chen, A. Utilizing the Variability of Resistive Random Access Memory to Implement Reconfigurable Physical Unclonable Functions. *IEEE Electron Device Lett.* **2015**, *36*, 138–140. [CrossRef]
46. Govindaraj, R.; Ghosh, S.; Katkooi, S. Design, Analysis and Application of Embedded Resistive RAM Based Strong Arbiter PUF. *IEEE Trans. Dependable Secur. Comput.* **2020**, *17*, 1232–1242. [CrossRef]

47. Mathew, J.; Chakraborty, R.S.; Sahoo, D.P.; Yang, Y.; Pradhan, D.K. A Novel Memristor-Based Hardware Security Primitive. *ACM Trans. Embed. Comput. Syst.* **2015**, *14*, 1–20. [\[CrossRef\]](#)
48. Chatterjee, U.; Chakraborty, R.S.; Mathew, J.; Pradhan, D.K. Memristor Based Arbiter PUF: Cryptanalysis Threat and Its Mitigation. In Proceedings of the 2016 29th International Conference on VLSI Design and 2016 15th International Conference on Embedded Systems (VLSID), Kolkata, India, 4–8 January 2016; pp. 535–540. [\[CrossRef\]](#)
49. Uddin, M.; Majumder, M.B.; Beckmann, K.; Manem, H.; Alamgir, Z.; Cady, N.C.; Rose, G.S. Design Considerations for Memristive Crossbar Physical Unclonable Functions. *ACM J. Emerg. Technol. Comput. Syst.* **2018**, *14*, 1–23. [\[CrossRef\]](#)
50. Ibrahim, H.M.; Abunahla, H.; Mohammad, B.; AlKhazaimi, H. Memristor-based PUF for lightweight cryptographic randomness. *Sci. Rep.* **2022**, *12*, 8633. [\[CrossRef\]](#) [\[PubMed\]](#)
51. Al-Tamimi, A.; Ali, S.; Cao, Y.; Bermak, A. Threshold Voltage based Dual Memristor Crossbar PUF. *AEU—Int. J. Electron. Commun.* **2024**, *175*, 155012. [\[CrossRef\]](#)
52. Rana, M.; Mamun, Q.; Islam, R. Lightweight cryptography in IoT networks: A survey. *Future Gener. Comput. Syst.* **2022**, *129*, 77–89. [\[CrossRef\]](#)
53. Xue, X.; Wang, C.; Liu, W.; Lv, H.; Wang, M.; Zeng, X. An RISC-V Processor with Area-Efficient Memristor-Based In-Memory Computing for Hash Algorithm in Blockchain Applications. *Micromachines* **2019**, *10*, 541. [\[CrossRef\]](#)
54. Ebrahimi, S.; Bayat-Sarmadi, S. Lightweight and DPA-Resistant Post-Quantum Cryptoprocessor based on Binary Ring-LWE. In Proceedings of the 2020 20th International Symposium on Computer Architecture and Digital Systems (CADS), Rasht, Iran, 19–20 August 2020; pp. 1–6. [\[CrossRef\]](#)
55. Matsuda, K.; Fujii, T.; Shoji, N.; Sugawara, T.; Sakiyama, K.; Hayashi, Y.I.; Nagata, M.; Miura, N. A 286 F²/Cell Distributed Bulk-Current Sensor and Secure Flush Code Eraser Against Laser Fault Injection Attack on Cryptographic Processor. *IEEE J. Solid-State Circuits* **2018**, *53*, 3174–3182. [\[CrossRef\]](#)
56. Lederer, I.; Mayer, R.; Rauber, A. Identifying Appropriate Intellectual Property Protection Mechanisms for Machine Learning Models: A Systematization of Watermarking, Fingerprinting, Model Access, and Attacks. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, 1–19. [\[CrossRef\]](#) [\[PubMed\]](#)
57. Zou, M.; Zhou, J.; Cui, X.; Wang, W.; Kvatinisky, S. Enhancing Security of Memristor Computing System through Secure Weight Mapping. In Proceedings of the 2022 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Nicosia, Cyprus, 4–6 July 2022. [\[CrossRef\]](#)
58. Lin, N.; Chen, X.; Lu, H.; Li, X. Chaotic Weights: A Novel Approach to Protect Intellectual Property of Deep Neural Networks. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2021**, *40*, 1327–1339. [\[CrossRef\]](#)
59. Cai, Y.; Chen, X.; Tian, L.; Wang, Y.; Yang, H. Enabling Secure NVM-Based in-Memory Neural Network Computing by Sparse Fast Gradient Encryption. *IEEE Trans. Comput.* **2020**, *69*, 1596–1610. [\[CrossRef\]](#)
60. Yasin, M.; Sinanoglu, O. Evolution of logic locking. In Proceedings of the 2017 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC), Abu Dhabi, United Arab Emirates, 23–25 October 2017; pp. 1–6. [\[CrossRef\]](#)
61. Boenisch, F. A Systematic Review on Model Watermarking for Neural Networks. *Front. Big Data* **2021**, *4*, 729663. [\[CrossRef\]](#) [\[PubMed\]](#)
62. Chen, C.Y.; Chakrabarty, K. On-line Functional Testing of Memristor-mapped Deep Neural Networks using Backdoored Checksums. In Proceedings of the 2021 IEEE International Test Conference (ITC), Anaheim, CA, USA, 10–15 October 2021; pp. 83–92. [\[CrossRef\]](#)
63. Sehra, K.; Singh, R.P.; Singh, S.; Wadhera, S.; Kasturi, P.; Saxena, G.J.; Saxena, M. Secure Digital Image Watermarking Using Memristor-Based Hyperchaotic Circuit. 2022. Available online: <https://link.springer.com/article/10.1007/s00371-022-02601-3> (accessed on 19 September 2023).
64. Chang, C.-Y.; Su, S.-J. A Neural-Network-Based Robust Watermarking Scheme. In Proceedings of the 2005 IEEE International Conference on Systems, Man and Cybernetics, Waikoloa, HI, USA, 10–12 October 2005; Volume 3, pp. 2482–2487. [\[CrossRef\]](#)
65. Huang, S.; Zhang, W.; Feng, W.; Yang, H. Blind watermarking scheme based on neural network. In Proceedings of the 2008 7th World Congress on Intelligent Control and Automation, Chongqing, China, 25–27 June 2008; pp. 5985–5989. [\[CrossRef\]](#)
66. Zhao, J.; Madduri, S.; Vadlamani, R.; Burleson, W.; Tessier, R. A Dedicated Monitoring Infrastructure for Multicore Processors. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2011**, *19*, 1011–1022. [\[CrossRef\]](#)
67. Chen, C.Y.; Chakrabarty, K. Pruning of Deep Neural Networks for Fault-Tolerant Memristor-based Accelerators. In Proceedings of the 2021 58th ACM/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 5–9 December 2021; pp. 889–894. [\[CrossRef\]](#)
68. Sun, X.; Yu, S. Impact of Non-Ideal Characteristics of Resistive Synaptic Devices on Implementing Convolutional Neural Networks. *IEEE J. Emerg. Sel. Top. Circuits Syst.* **2019**, *9*, 570–579. [\[CrossRef\]](#)
69. Chen, C.Y.; Chakrabarty, K. Efficient Identification of Critical Faults in Memristor-Based Inferencing Accelerators. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2022**, *41*, 2301–2314. [\[CrossRef\]](#)
70. Ltd, A. A Chiplet Marketplace for Next-Generation Performance. *Arm | The Architecture for the Digital World*. Available online: <https://www.arm.com/markets/technology-topics/chiplets> (accessed on 25 February 2024).
71. Intellectual Property Crime Threat Assessment 2022. *Europol*. 2022. Available online: <https://www.europol.europa.eu/publications-events/publications/intellectual-property-crime-threat-assessment-2022> (accessed on 22 September 2023).

72. Mutschler, A. Chiplet Security Risks Underestimated. *Semiconductor Engineering*. 2023. Available online: <https://semiengineering.com/chiplet-security-risks-underestimated/> (accessed on 25 February 2024).
73. Lin, L.; Burleson, W.; Paar, C. MOLES: Malicious off-chip leakage enabled by side-channels. In Proceedings of the ICCAD'09: The International Conference on Computer-Aided Design, San Jose, CA, USA, 2–5 November 2009; pp. 117–122. [CrossRef]
74. Kim, J.; Chekuri, V.C.K.; Rahman, N.M.; Dolatsara, M.A.; Torun, H.M.; Swaminathan, M.; Mukhopadhyay, S.; Lim, S.K. Chiplet/Interposer Co-Design for Power Delivery Network Optimization in Heterogeneous 2.5-D ICs. *IEEE Trans. Components, Packag. Manuf. Technol.* **2021**, *11*, 2148–2157. [CrossRef]
75. Vashistha, N.; Al Hasan, M.M.; Asadizanjani, N.; Rahman, F.; Tehranipoor, M. Trust Validation of Chiplets using a Physical Inspection based Certification Authority. In Proceedings of the 2022 IEEE 72nd Electronic Components and Technology Conference (ECTC), San Diego, CA, USA, 31 May–3 June 2022; pp. 2311–2320. [CrossRef]
76. Mosavirik, T.; Schaumont, P.; Tajik, S. ImpedanceVerif: On-Chip Impedance Sensing for System-Level Tampering Detection. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2022**, *2023*, 301–325. [CrossRef]
77. Deric, A.; Holcomb, D. Know Time to Die - Integrity Checking for Zero Trust Chiplet-based Systems Using between-Die Delay PUFs. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2022**, *2023*, 391–412. [CrossRef]
78. Sperling, E. Security Risks Widen with Commercial Chiplets. *Semiconductor Engineering*. 2022. Available online: <https://semiengineering.com/security-risks-widen-with-commercial-chiplets/> (accessed on 25 September 2023).
79. Xu, X.; Rahman, F.; Shakya, B.; Vassilev, A.; Forte, D.; Tehranipoor, M. Electronics Supply Chain Integrity Enabled by Blockchain. *ACM Trans. Des. Autom. Electron. Syst.* **2019**, *24*, 1–25. [CrossRef] [PubMed]
80. Das Sharma, D. Universal Chiplet Interconnect Express (UCIe)®: Building an Open Chiplet Ecosystem. 2022. p. 7. Available online: <https://www.uciexpress.org/ucie-1-0-white-paper-download> (accessed on 25 September 2023).
81. Drucker, K.; Jani, D.; Agarwal, I.; Miller, G.; Mittal, M.; Wang, R.; Vinnakota, B. The Open Domain-Specific Architecture. In Proceedings of the 2020 IEEE Symposium on High-Performance Interconnects (HOTI), Piscataway, NJ, USA, 19–21 August 2020; pp. 25–32. [CrossRef]
82. Das Sharma, D.; Agarwal, I. Compute Express Link 3.0. p. 4. Available online: https://computeexpresslink.org/wp-content/uploads/2023/12/CXL_3.0_white-paper_FINAL.pdf (accessed on 25 September 2023).
83. Holcomb, D.E.; Fu, K. Bitline PUF: Building Native Challenge-Response PUF Capability into Any SRAM. In Proceedings of the 16th International Workshop, Busan, Republic of Korea, 23–26 September 2014; Series Title: Lecture Notes in Computer Science; Volume 7908, pp. 510–526. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.