6-2023

A survey on security analysis of machine learning-oriented hardware and software intellectual property

Ashraful Tauhid

Lei Xu

Mostafizur Rahman

Emmett Tomai

High-Confidence Computing 3 (2023) 100114

Contents lists available at ScienceDirect

High-Confidence Computing

journal homepage: www.sciencedirect.com/journal/high-confidence-computing

Review

A survey on security analysis of machine learning-oriented hardware and software intellectual property

Ashraful Tauhid [a,*], Lei Xu [b], Mostafizur Rahman [c], Emmett Tomai [a]

[a] Department of Computer Science, University of Texas Rio Grande Valley, TX 78539, USA
[b] Department of Computer Science, Kent State University, OH 44240, USA
[c] Department of Computing and Mathematics, University of West Georgia, GA 30118, USA

ARTICLE INFO

Article history:
Received 7 November 2022
Revised 8 January 2023
Accepted 26 January 2023

Keywords:
Intellectual property
IP protection
Patent
Copyright
Trademark
Infringement
Machine learning
Integrated circuit

ABSTRACT

Intellectual Property (IP) includes ideas, innovations, methodologies, works of authorship (viz., literary and artistic works), emblems, brands, images, etc. This property is intangible since it is pertinent to the human intellect. Therefore, IP entities are indisputably vulnerable to infringements and modifications without the owner's consent. IP protection regulations have been deployed and are still in practice, including patents, copyrights, contracts, trademarks, trade secrets, etc., to address these challenges. Unfortunately, these protections are insufficient to keep IP entities from being changed or stolen without permission. As for this, some IPs require hardware IP protection mechanisms, and others require software IP protection techniques. To secure these IPs, researchers have explored the domain of Intellectual Property Protection (IPP) using different approaches. In this paper, we discuss the existing IP rights and concurrent breakthroughs in the field of IPP research; provide discussions on hardware IP and software IP attacks and defense techniques; summarize different applications of IP protection; and lastly, identify the challenges and future research prospects in hardware and software IP security.

1. Introduction

Intellectual Property (IP) refers to anything that originates in the human mind, including theories, conceptions, discoveries, anecdotes, works of literature, etc. IP has a few features, which include the uniqueness of creation and application, the portrayal of the author's personality, and, nonetheless, no obstruction or interference from any third party while exercising the exclusive rights of the IP owner [1]. To identify the uniqueness and authorship of these properties, it is of the essence that the rights to protect the IPs are in practice. Protecting IP is related to how it can be used and practiced under lawful rights. Plagiarism, covert exploitation of intellectual effort and intellectual theft are all examples of IP infringements. As a countermeasure to these IP infringements, a set of defined regulations, Intellectual Property Rights (IPR), are put into practice. Of all the IPRs, industrial designs, patents, trade secrets, copyrights, geographical markers, layout designs for Integrated Circuits (IC), and trademarks are some of the most prevalent forms [2]. While IPR proposes more of a legal framework to defend the IPs from infringement by implying consequences for infringement, Intellectual Property Protection (IPP) is concerned with how the IPs can be protected within the medium, whether it is software-based or hardware-based. Hence, IPP can actively resist and deny the chances of getting the IPs infringed, whereas IPR passively controls the infringement of IPs.

In today's economy, robust IP protection plays a vital role in attracting multinational companies from advanced countries to launch their businesses in other parts of the world (especially in developing countries) without any hesitance [3]. Unfortunately, the existing laws and regulations cannot stop the unlawful exercise of IP because of its passive nature. Consequently, inadequate IP protection techniques have resulted in numerous vulnerabilities ranging from reverse engineering and piracy to hardware trojans [4–6]. To address these threats and to secure the integrity of hardware IP, IPP offers various generic techniques, among which hardware obfuscation [7–10], hardware metering [11–14], split manufacturing [15–19], logic locking [5,20,21], fingerprinting & watermarking [22–25] and IC camouflaging [26–33] are noteworthy. For software IP protection, different approaches are already in practice, among which the unique techniques are fingerprinting and watermarking [34–37], steganographic techniques [38–42], code obfuscation [43], and surprisal analysis [44].

The use of Machine Learning (ML) is the second and most recent alternative to protect IP paradigms apart from the generic approaches. Although machine learning theoretically resides more on the software edge of a computing system, it can nevertheless be used to protect hardware IPs. The premise underlying

* Corresponding author.
E-mail address: ashtauhid@gmail.com (A. Tauhid).

https://doi.org/10.1016/j.hcc.2023.100114

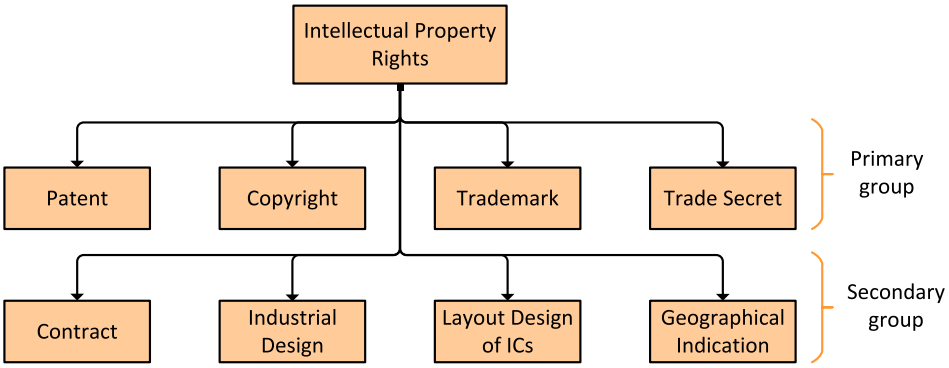

Fig. 1. Types of Intellectual Property Rights.

the protection of hardware IP using ML is that, if properly trained, ML models can identify any slight change in hardware behavior. Different ML-based techniques are proven to be resistant against hardware trojan attacks [45–50], side-channel attacks [51,52], IC counterfeit/reverse engineering attacks [53,54], and attacks on Physical Unclonable Functions (PUF) [55–59]. The features on which these models primarily focus are power usage, latency in run-time, electromagnetic emission data, memory access pattern, current supply, aging deterioration of recycled ICs etc. Additionally, among other ML techniques, Convolutional Neural Networks (CNN) [60,61], Generative Adversarial Network (GAN) [62], and Deep Neural Networks (DNN) [63] are used to secure software IPs against fault-injection attacks, adversarial attacks, trojan attacks, backdoor attacks, model inversion attacks, black-box attacks, and man-in-the-middle attacks.

Our understanding of prior IPP research indicates that hardware intellectual property and software intellectual property has been covered independently in several papers. However, they have not been brought under the same umbrella and discussed as a stark scenario. Hence, in contrast to the existing literature, this paper looks at these topics from a generic perspective and a machine learning point of view to mitigating the attacks.

The rest of the paper is structured as follows: A contrast between IPR and IPP, including how they work, is discussed in Section 2, followed by the IPP attack and defense strategies covered in Section 3. Next, Section 4 lists a few noteworthy IP protection applications. Finally, Section 5 discusses the challenges and future work relevant to IPP, and Section 6 discuss the concluding remarks and the author's competing interests, respectively.

2. Intellectual property rights vs. intellectual property protection

2.1. Intellectual Property Rights

Intellectual Property Rights act as a legal protection against IP infringements, and they can be broadly classified into four categories: Patent, Copyright, Trademark, and Trade Secret. Other crucial IPRs include geographical indications, industrial design, and IC layout design. Fig. 1 provides a graphical view in organizing IP rights into primary and secondary groups.

Patent. A patent is an inventor's lawful right to prevent others from copying, recreating, or exploiting his/her innovation. This privilege also called an "Intellectual Property Right", is widely regarded as one of the most important driving forces behind ingenuity and innovation. In 1718, Britain introduced documentation and design of the innovation as a prominent part of the patent, which is still in practice in most countries [64].

Copyright. Copyright is defined under the laws as a form of protection of authorship for authentic and original works in a tangible form, which includes computer programs, movies, databases, photos, musical compositions, sculptures, etc. Since many of these IPs are in intangible form, these works cannot be protected by copyright or patent; therefore, copyright enforces a tangible form to protect them [65].

Contract. Contracts can be used to create, protect, or transfer IP rights to another party. However, contracts must meet at least one of these conditions to be an IP: confidentiality or non-disclosure agreements, agreements transferring IP, license agreements, joint development agreements, and optional agreements. According to the theoretical model proposed by Harris et al. [66], novice IP creators should be provided with a different type of contract than experienced creators.

Trademark. A trademark is a distinctive symbol, logo, design, or expression that can be used to distinguish the origin of one's goods or services from those of another. Characters, drawings, words, pictures, symbols, numbers, and even acoustics may be used to create a trademark. Trademark protection is concerned with the consumer's preferences or with aiding the consumer's welfare [67].

Trade Secret. Unlike patents, copyrights, and trademarks, modern trade secrets were practiced during the nineteenth century [68]. Marketing practices, logistic operations, client data, promotional tactics, vendor and consumer lists, and production techniques are all examples of trade secrets.

Industrial Design. Industrial design includes the shape, composition, color, pattern, configuration, and decorative or aesthetic design of products that are valuable to the customer or the producer. In the US, industrial design falls under any of these categories: protection accorded by laws governing patents, legal protection under copyright, and protection accorded by the law governing trademarks [69].

Layout Design of ICs. Because of the rising expense and complexity of semiconductor fabrication, many firms are compelled to outsource their designs and productions, exposing them to trojans installed in offshore factories, unauthorized and surplus production of ICs, and IP piracy [70]. Hence, IPs that are related to the layout design of ICs are protected by IPR laws.

Geographical Indications. The term Geographical Indication (GI) refers to the fact that a product originates from a specific location and possesses distinctive traits attributable to that location. Therefore, it also falls under the umbrella of IPR.

2.2. Intellectual Property Protection

Hardware IPP. Protection of IP can be implemented into a system in two distinct ways: through hardware and software. The focus of hardware IPP is on how it can be protected from

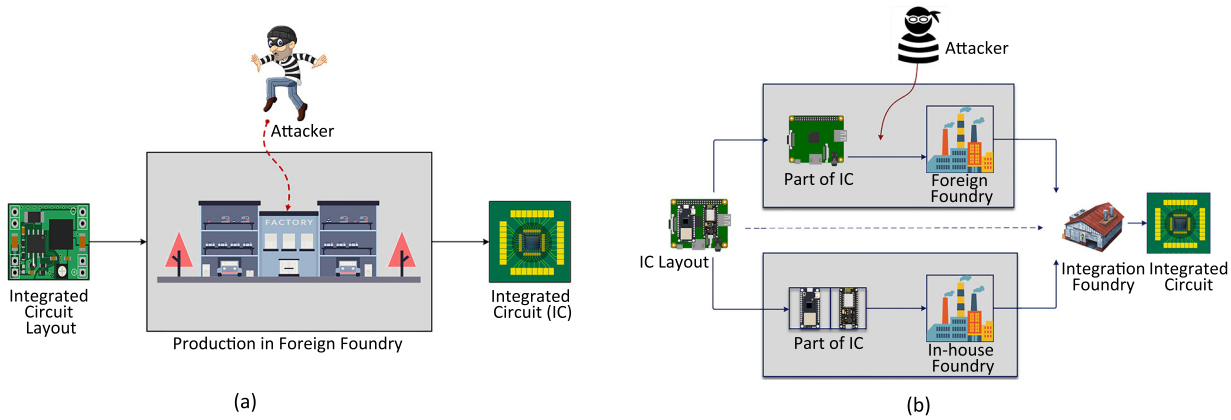

Fig. 2. Hardware IPP. (a) A general overview of how a hardvare IP attack works. (b) Split manufacturing IPP.

the foundry onward, whereas in most cases, software IPP can be deployed at any point, even after the system has been operational. For example, hardware IPP is often implemented during the IC manufacturing process, and after it has been distributed to end users, it either cannot be changed or any changes would require a significant investment of time and money. Software IPPs, as opposed to hardware IPPs, can be modified later via an updated patch. Both of these IPP configurations have vulnerabilities that can be exploited against them. Fig. 2(a) provides a general illustration of how most of the hardware IPs are compromised by intruders.

The IC design yielded within the confines of the company is represented by the left side of the figure. This is the IP component of the company that needs to be protected. This design is fed into an off-shore factory, which is depicted in the middle of 2(a). Since most enterprises depend on chips that are outsourced from a foreign foundry, adversaries can be effortlessly introduced into the chips without resulting in any or just a minor alteration to their logical or physical form. This opens the door for an attacker to exploit the IPs. The most challenging aspect is that it is challenging to identify the adversary operating in a low-profile setting. In response to this situation, researchers have proposed several different strategies, such as segmenting the entire IC fabrication process so that unauthorized individuals cannot get a glimpse of the entire design. They facilitated this by adding unique IDs to the chips, incorporating dummy contacts, fingerprinting, watermarking, etc. In addition, a famous defense technology called split manufacturing works efficiently to protect hardware IPs. This is depicted in Fig. 2(b) where the IC layout is the company's IP. It is divided into two segments and sent to two different foundries. Each foundry fabricates a portion of the IC and, finally, they are integrated at the integration foundry. Since one foundry does not have the overall architecture of the IC, it is extremely difficult to incorporate a functioning adversary without hampering the design and efficacy of the chip. There is one more threat in the process of Fig. 2(b), which is reverse engineering from the integration house. Logic locking, IC camouflaging, hardware obfuscation, or hardware metering are viable measures for this. Aside from these measures, neural networks are seen to be used recently on "Edge ML" devices to protect against hardware-based attacks.

Software IPP. Software IPs are starkly different from hardware IPs. Current technology allows for multiple ways to access, modify, and fabricate these IP entities. When it comes to software IP attacks, machine learning models are especially defenseless. Numerous attacks can be launched against a model, including fault tolerance attacks, adversarial attacks, model inversion attacks, neural trojan attacks, backdoor attacks, etc. Fig. 3 depicts a standard neural network model with an input layer that takes the input to the model, hidden layers that perform the model's underlying calculations, and an output layer that either classifies, predicts, or clusters the inputs to different categories. Therefore, for an adversary to successfully disrupt the operations of a neural network, it is necessary to either manipulate the input or corrupt the interim calculations of the model. A typical attack can be made on the model's weight parameters or the bias. In [71], the authors presented a method that allows the weight values of the DNN model to be altered, which has the potential to influence the results produced by the model. For instance, Fig. 3 has an image of a cat as input, which the model classifies as a cat under the normal scenario. However, if an attacker changes a weight parameter, the model might conclude that the image is of a car or a tennis ball. This happens because if a weight value is changed in one of the hidden layers, it will reflect on the rest of the layers through propagation in the network, and as a result, it is extremely likely to generate an inaccurate output. Modifying the network's inputs is another example of a traditional attack on neural networks. In [72], the authors conducted an experiment in which they tried sticking a post-it note to a stop sign. Consequently, the neural network model consistently and deliberately misinterpreted it as a sign indicating the speed limit. This shows how the application field of neural networks might be affected by faulty inputs or just a minor weight parameter modification of the network. Therefore, overcoming these challenges is crucial to use software IPs like artificial intelligence and machine learning in real-world situations. In addition to the machine learning models, traditional software is also vulnerable to IP infringements. So, this software also needs to be protected using software IPP schemes.

3. Review on attacks and protection schemes of IP

Hardware and software are both integral components of a digital system. It is, therefore, critical to secure both hardware and software to ensure the system's total confidentiality, integrity, and availability. This is why the security of IPs against several potential attacks has been investigated using a plethora of techniques for both hardware and software. According to the existing literature based on those techniques, this section describes IPs' available attack and defense strategies. The organization of section is divided into four sub-sections. The first two sub-sections discuss the attack and protection strategies of hardware IP, and the last two sub-sections discuss software IP's attack and protection schemes.

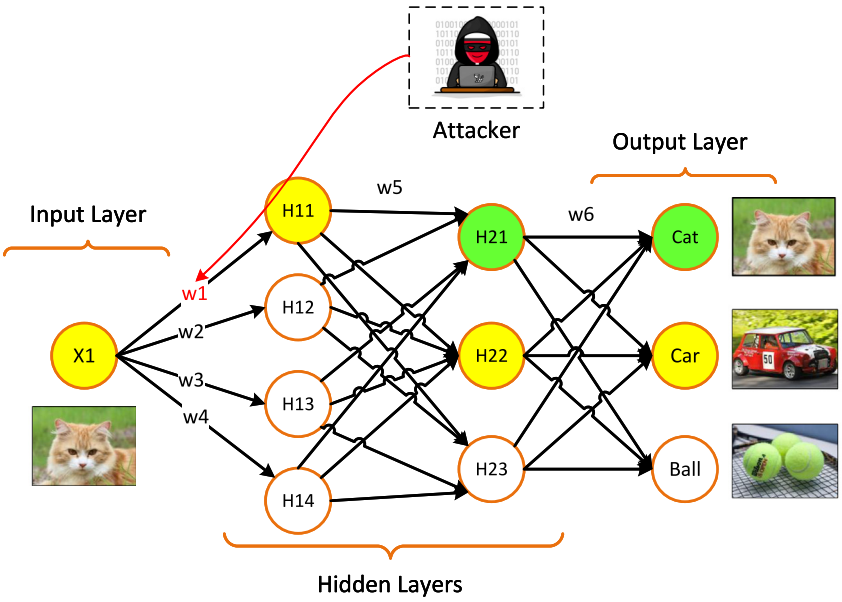

Fig. 3. Attack on weight parameters of a neural network.

3.1. Attacks and Protection Schemes of Hardware IP

Protection of hardware is often necessary to preserve confidentiality, integrity, and availability (CIA) of asset(s) built into hardware. However, on many occasions, those security properties of asset have been broken. Accordingly, defense against those attacks have been proposed. We discuss some of the prominent attacks on hardware assets and corresponding defenses in this section.

3.1.1. Attacks

Hardware Trojan. A hardware trojan is any malicious hardware alteration, regardless of how innocuous it appears to be, that opens a backdoor for an attacker to break into the system [73–75]. In the past decade, it has been concerning issue for manufacturers who outsource any of their components to an outside foundry [76].

Reverse Engineering. Learning about a system through its constituent elements' methodical analysis (parameter analysis, output analysis, input–output relationship) is known as reverse engineering [77]. This strategy ranks among the most common ones used in hardware and software IP attacks.

Unauthorized Production. Another adversary that can compromise the integrity of hardware security is an overproduction or unauthorized production. This happens because of the lack of monitoring over a remote foundry while outsourcing the hardware (mostly ICs) [78–80]. Unauthorized productions help the intruders to examine the ICs and reverse engineer them or sometimes install the adversary itself into the hardware [81].

Sensitization Attack. This attack is most prominent in the logic-locking configuration of hardware IP protection. The process of incorporating more logic into a circuit while simultaneously preventing alterations to the preliminary design with a secret key is called logic locking. Sensitization attack makes the essential confidential key bits of the algorithms vulnerable to attack by sensitizing them. There are three primarily main methods of logic locking: Random Logic Locking (RLL), Fault analysis-based Logic Locking (FLL), and Strong Logic Locking (SLL). RLL and FLL, in contrast to SLL, can be easily broken by this technique [82].

SAT Attack. The SAT attack, also known as a boolean satisfiability-based attack, is a powerful method to defeat all current logic-locking schemes, including SLL. This attack is built on the concept of incorrect-key removal via distinguishing input patterns (DIPs) [83].

Decamouflaging Attack. The decamouflaging attack is closely linked to the IC camouflaging technique of hardware IPs. IC camouflaging is manufactured with a combination of filler cells [26], dummy contacts [84], and standard programmable cells [85] etc. When reverse engineering is performed on a camouflaged IC, the IC becomes decamouflaged for a particular group of input patterns [86,87].

Proximity Attack. Proximity attacks can be used in a split manufacturing approach, in which the task of manufacturing ICs is divided into two layers: back-end-of-line (BEOL) and front-end-of-line (FEOL). This attack retrieves BEOL connections from the FEOL layers using heuristics of the physical design [18].

These attack techniques are devised for different circumstances that vary in complexity and extent of damage to the respective IPs. Based on the detrimental impacts on the IP entities, we can rank them according to the Table 1.

Table 1
Indexing of hardware IP attacks.

Index	Attack name
1	Hardware Trojan
2	Reverse Engineering
3	Unauthorized Production
4	Sensitization Attack, SAT Attack, Decamouflaging Attack, Proximity Attack

With time the attacking approach has evolved from simply implanting hardware trojans in ICs to targeting a much wider spectrum of hardware components [88]. Also, the proliferation of trojans at multiple stages of production is a consequence of increasingly complex systems [89] which increases the chances of jeopardizing the IPs through hardware trojans. These two factors are primarily responsible for a hardware's root-of-trust being compromised. Hence, hardware trojan is the most challenging attack to defend against. For reverse engineering, despite its prevalence for decades, the assessment of its sophistication remains unresolved since the complexity arises from both technological and psychological perspectives [90]. Hence, this attack is indexed as the second most effective attack strategy. Unauthorized production is not a technical attack strategy but a supply-chain-based attack [81]. So, the severity is less than the hardware trojan

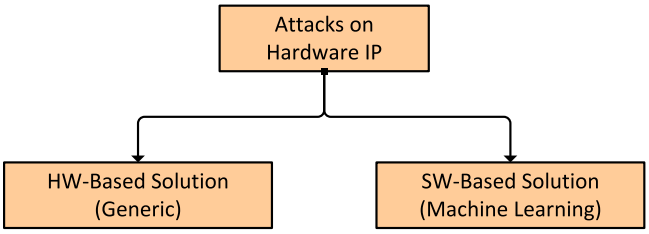

Fig. 4. Defense strategies against hardware-IP attacks.

and reverse engineering as it can be monitored, examined, and protected anytime. An IC's netlist, which details the connectivity and functionality of the circuit, is the target of sensitization attacks. This is essentially limited to logic-locking contexts [82]. SAT attacks are mostly relevant to hardware IP protection that enables logic locking. Similarly, decamouflaging attacks are limited to only attacking the camouflaged ICs and proximity attacks target the split manufacturing configurations.

3.1.2. Protection Schemes

ICs are vulnerable to security risks because of the worldwide and scattered semiconductor supply chain. Dedicated hardware root-of-trust is included in several commercial devices for completing safety functions. However, even though it can protect the digital circuit, it cannot erase the circuit's vulnerabilities. There are two prevalent solutions to these kind of attacks. One is the generic protection approach, and the other is the machine learning model-based solution (see Fig. 4). With appropriate protection schemes, intrusion can be detected both online and offline.

Generic Approaches. The major generic hardware-oriented approaches to defending against hardware-IP attacks include Hardware Metering/IC Metering, Hardware Obfuscation, Split Manufacturing, IC Camouflaging, Logic Locking, Fingerprinting, and Watermarking. These strategies are used in different need-based scenarios. For example, split manufacturing is not applicable if one foundry produces the entire chip. Instead, IC camouflaging can be used to protect the IP of the chips within the confine of the foundry. Table 2 summarizes these protection strategies against attacks for generic hardware.

(i) Hardware Metering/IC Metering: Hardware metering as an IP protection scheme for ICs was first proposed by Koushanfar et al. in [11]. They inserted unique identifiers (IDs) into the ICs and demonstrated that distinct IDs might cover many chips. It provides the IC designers the ability to lock individual IC, which could control IP piracy & overproduction. The problem with this technique is that it uses the idea of passive metering. Alkabani et al. [12] proposed an active metering where unclonable variability-based IDs were designed, which could lock the ICs at the foundry. They generated Boosted Finite State Machines (BFSM) by including additional states and transitions to the original Finite State Machine (FSM). Subsequently, they incorporated black hole states into the design of BFSM to suspend individual ICs remotely. Alkabani et al. [13] proposed another active metering scheme where they replicated a few FSM states and deployed dynamic IDs to lock the ICs, unlike in [12] which deployed static IDs.

(ii) Hardware Obfuscation: The term "hardware obfuscation" describes making the circuit structure and operation obscure and challenging to understand to significantly increase the expense of reverse engineering [7,8]. Li et al. [9] investigated a gate-level obfuscation technique that employed a PUF and programmable logic to hide the true nature of logic operations. Alaql et al. [10] introduced LeGO, a learning-guided obfuscation framework. This framework analyzes obfuscated IP to determine whether an attack has occurred or not and identifies the vulnerability. Then, to remove this vulnerability, a set of design adjustment steps are performed on this model. In a white-box context, Chakraborty et al. [94] provided a simple concealment scheme named Hardware Protected Neural Network (HPNN) to ensure that DL models meet the necessary IP security requirements. This algorithm's security depends on using a "hardware root-of-trust" that incorporates a private key into on-chip memory. By storing the secret key on the chip, this architecture ensures that only the end user with a reliable hardware device can run the intended DL applications. Goldstein et al. [95] proposed a different type of hardware obfuscation strategy that can conceal the native network's underlying structure by exchanging rows and columns based on a covert mapping key to prevent IP theft in edge devices and unauthorized use of DL models during the dissemination, installation, and operation phases.

(iii) Split Manufacturing: Most IC design companies now operate with a fabless strategy of production and supply chain of ICs because of the high costs associated with operating a manufacturing site. Split manufacturing was devised to accommodate the fabrication and distribution of ICs without a facility. In split manufacturing, ICs are produced using one or more metal layers called MX between the top and bottom layers. Vaidyanathan et al. [16] proposed a secure split fabrication of ICs using the lowest metal layer called metal1 (M1). Until now, M1 has been considered the most secure layer for preventing the unauthorized production of ICs [96]. But one major problem with M1 is that it incurs high cost of production [17]. Another good strategy of split manufacturing is to add dummy connections and cells to the IC design. It can significantly impact resolving reliability, scalability, and privacy issues, as well as defend against hardware Trojan and counterfeiting attacks. In light of this concept, Li et al. [19] offered a resolution to the privacy considerations and computational demands of split manufacturing using a graph-based approach. They added dummy connections and cells within the ICs by employing a technique known as Mixed-Integer Linear Programming (MILP), which led to improved results (better privacy, better efficacy, and low overhead) according to the simulation. Like this strategy, Xiao et al. [17] proposed adding an auxiliary operational circuit called Obfuscated Built-in Self-Authentication (OBISA) and a higher metal layer in BEOL. The metal layer is expected to be greater than three because metal layers lower than 4 require higher production costs.

(iv) IC Camouflaging: To obfuscate the accurate logic operations of a circuit from visual inspection, researchers have developed a technique known as "circuit camouflage". This obfuscation includes redesigning cells, masking circuits, and remodeling the conventional architecture of ICs. Rajendran et al. in [27] proposed circuit camouflage, a form of hardware camouflage, to safeguard valuable IPs from counterfeiting attacks and hardware trojan attacks. They camouflaged only the crucial IC components, which decreased area and computational overhead while providing adequate resilience against IP attacks. Another approach to integrating circuit camouflage in ICs was proposed in [28], suggesting populating empty areas in an IC layout to prevent an attacker from installing trojans. Also, this can be protected by implementing a substantial number of camouflaging gates because the difficulty of finding the proper

Table 2
Generic: Attack and defense strategies in hardware IP.

Attacks	Protection strategy
Malicious Collusion, Reverse Engineering	Fingerprinting & Watermarking [22–24]
Overproduction	Hardware Metering [11]
IP Piracy, Run-time Tempering	Hardware Metering/IC Metering [12,13]
Unauthorized Production	IC Locking [14]
IP Piracy, Reverse Engineering	Logic Locking [5,91]
Circuit Re-Synthesis, State Reduction Attack	Fingerprinting & Watermarking [25]
Reverse Engineering, Hardware Trojan	IC Camouflaging [27–29], Split Manufacturing [17–19]
Unauthorized Production	Split Manufacturing [16]
Reverse Engineering, IC Design Piracy	Hardware Obfuscation [9,10]
Counterfeiting, Hardware Trojan, Overproduction	Logic Locking [92,93]
Reverse Engineering	IC Camouflaging [33]
Unauthorized Use of DL Model	Hardware Obfuscation [94,95]

inputs is significantly correlated to the number of camouflaging gates being used [97]. However, this introduces a significant overhead due to numerous dummy contacts. To address these challenges, researchers are now using camouflaging techniques that are based on the threshold voltage value [29–32]. Besides, to hide cells from existing imaging tools that are used to reverse engineer the ICs, authors in [33] created a technique called "covert gate" that takes advantage of fabricated connections to produce cells that appear entirely normal.

(v) Logic Locking: Like the name suggests logic locking delivers IP security by locking a design using a secret key [21,98–100]. Potluri et al. [91] proposed SeqL, an IP Protection with Secure Scan-Locking. Seql performs FI-SQ locking and functional isolation. As a result, the decrypted key is no longer practical. The attacker cannot see functionally valid keys since SeqL hides them, and the chances of being functionally wrong about the decrypted key are increased. Roy et al. [5] facilitated a chip-locking and unlocking framework using public key cryptography. According to their methodology, each IC has a unique, unclonable external token that only the IP owner can issue. Without the tokens, the chips are designed not to operate correctly. In terms of SAT attack, Yasin et al. [92] proposed SARLock and Xie & Srivastava [93] proposed Anti-SAT. SARLock works with a hard-coded key value of the logic gate, which is used to corrupt all outputs except the correct ones, whereas Anti-SAT works with a new block with inputs converging to an AND gate in addition to the existing logic locking circuit. For Anti-SAT, the key is correct if the AND gate outputs 0; otherwise, it is incorrect.

(vi) Fingerprinting and Watermarking: Lach et al. [22] proposed a hardware IP protection algorithm based on fingerprinting techniques for Field Programmable Gate Arrays (FPGA). Another paper published by the same authors discusses a technique to use watermarking in the physical layout of an FPGA [23]. Kahng et al. [24] presented a new watermarking that is practically imperceptible to human eyes which makes it challenging to overwrite or remove, and inextricably integrated into the design. Static fingerprinting is most prevalent in all of these IP protection schemes. However, the first dynamic fingerprinting was proposed in [25], where watermarking was used to prove custody of a hardware IP, and the buyer's fingerprint served as a verification of their identity.

Machine Learning Based Approaches. Using machine learning models to find anomalies in hardware IP and techniques to secure it has recently gained popularity in hardware-IP security research. Machine learning is commonly used to address the following hardware-IP attacks: side-channel attacks, hardware trojan attacks, reverse engineering, IC overbuilding, IC counterfeiting, and attacks targeting the PUF of the ICs. Much research has been done on machine learning techniques to identify attacks on hardware IPs. Table 3 tabulates a list of detection mechanisms of hardware-IP invasion using machine learning models.

(i) Defense Against Hardware Trojan Attacks: Unusual behavior in any on-chip data sources is a primary indication that a hardware trojan has been activated. This can be measured at a microscopic level and used this data to train a trojan detection model. However, Most machine learning models are constructed to identify hardware trojans before going live. This opens the door for opportunities to deploy trojans after hardware becomes operational. Jin et al.et al. [45] figured out a way to repel this threat using a one-class neural network. In another work, Iwase et al. [46] proposed a similar technique using the SVM model, but unlike [45], their data was primarily collected from the frequency spectrum of power usage. Power usage was also used in [47] to analyze the patterns using a back-propagation neural network. Besides power usage, latency in run-time [48] and electromagnetic emission data [46,49] are also used to identify hardware trojans. Moreover, recent researches show that image processing [50] can also be used to identify hardware trojan.

(ii) Defense Against Side-Channel Attacks: An unobtrusive method of attacking hardware is side-channel analysis. It has no payload on the hardware because it infers statistical information from data points like power usage, run-time, current supply, memory access pattern, electromagnetic emission etc. All these data points are vulnerable to leakage because of the imminent nature of the hardware operation. Yan et al. [51] proposed a method to address side-channel attacks through memory access pattern. In order to prevent attackers from listening in on the user's memory access behavior, they recommended a robust hierarchical structure of cache replacement strategy. Wang et al. [52] proposed an AI model for the edge devices which has Hardware Performance Counter (HPC) registers built into the hardware. They asserted that with a sufficient detection rate this approach can withstand side-channel and malware attacks. However, the fact that this method needs an HPC register to function is a significant disadvantage.

(iii) Defense Against Reverse Engineering/IC Counterfeiting: Reverse engineering and IC counterfeiting are relatively the same concept that is to recycle from a hardware. This attack strategy is silent in nature because it is also unobtrusive method of attacking the IP. Huang et al. [53] proposed a machine learning strategy to counteract this attack by observing the aging deterioration of recycled IC in comparison to original IC. They used SVM to simulate their proposal and discovered that as few as 10 ICs are needed to identify the counterfeit ICs. Another similar work based on aging deterioration of ICs was proposed by Dogan et al. [54] using a 2-phase SVM classification.

Table 3
Machine learning based attack detection in hardware IP.

Attacks detection mechanisms	Machine learning techniques
IC Counterfeit	Support Vector Machine [53], Artificial Neural Network [101]
Reverse Engineering	Support Vector Machine [54,102]
Hardware Trojan	Support Vector Machine [46,102–105], Back-Propagation Neural Network [47], K-Means Clustering [106], Random Forest [107]
Rootkit	Support Vector Machine [108]
Neural Trojan on edge ML	Deep Neural Network [109]
Ransomware	Deep Neural Network [110]
Malware & Side-Channel Attack	Artificial Intelligence [52]

(iv) Defense Against Attacks on PUFs: When it comes to authenticating and identifying FPGA hardware, PUFs are essential for solving the key generation and storage challenges which were initially imposed by the volatile nature of memory in FPGA [59]. Assuming that PUF with unknown mathematical model can be contaminated with ML attacks, Ganji et al. [58] proposed an ML method to defend FPGA based Bistable Ring PUF (BR-PUF). Rührmair et al. [55] proposed an attack using logistic regression and evolution strategies on Ring Oscillator PUF (RO-PUF). A novel solution to this attack was proposed in [57] using Probably Approximately Correct (PAC) learning framework. Apart from machine learning approaches, a generic software-based hardware protection was proposed by Kohnhauser et al. [56]. They considered PUFs within the low-end embedded devices and deployed check-summing procedure of the code to establish the hardware protection.

Based on the generic techniques used to protect hardware IP, hardware obfuscation and IC camouflaging can be categorized as similar types of defensive strategies. These two techniques primarily confuse and deter the attacker with an overwhelming design. On the other hand, hardware/IC metering and logic locking can be categorized as another set of techniques that can address similar situations. Still, these techniques focus on seizing the capability of infringing the IP in the first place. Unlike the previous two techniques, split manufacturing is unique and is the most widely used technique in hardware IP security. Lastly, fingerprinting and watermarking are less used in hardware IP because of their over-writable traits. All of these state-of-the-art defense strategies provide protection pathway against conceivable hardware IP attack scenarios. However, it is impossible for any one of them to defend against all of the plausible threats. This is mostly due to the fact that there is no practical way to cluster all the threat models. So, when analyzing how well different defenses work, authors tend to look at one or more criteria and metrics at a time and overlook the others. In the case of hardware IPP based on machine learning, researchers typically concentrate on machine learning models to fend against threats of IP infringement or violation. The majority of the efforts have been addressed using SVM models, while a few have also used DNN, according to our reviewed articles. To further extend this research, new models can be tested against SVM and DNN to compare the results in addition to the hybrid models.

3.2. Attacks and Protection Schemes of Software IP

3.2.1. Attacks

Fault-Injection Attack. "Fault-injection attack" and "bit-flip attack" are two terms often used interchangeably because they both resemble the effects of changing bits inside of a DRAM [111]. By leveraging the fault-injection attack approach, Rakin et al. [71] developed an attack model for any DNN model wherein any slight modification to the weight bits of the DNN model stored in DRAM has an impact on the model's output. As a countermeasure to such bit-flip attacks, Li et al. [112] presented an innovative approach to weight reconstruction, in which the weights were reconstructed during inference in such a way as to limit or distribute the weight disturbance generated by BFA to the weights that were located close. The sensitivity of DNN against gradient-based and stochastic BFA fluctuations has been significantly improved using this technique. Even with the most vicious attacks (such as greedy bit search), after five iterations, this method still maintains a test accuracy of sixty percent on the ImageNet dataset.

Table 4
Indexing of software IP attacks.

Index	Attack name
1	Fault-Injection Attack, Trojan Attack
2	Geometric Attack

While these studies are largely concentrated on the offensive and defensive aspects of fault injection attacks, Javaheripi et al. [113] suggested a detection technique for this attack on neural networks. They used a hash function on the sensitive layers of the neural network model and compared them against the expected hash value. A mismatch in the two values implies that one or more weights have been altered. Unlike other attack models, Tatar et al. [114] proposed another bit-flip attack model that can initiate row-hammer bit-flips by sending merely network packets. This is an imminent threat to most cloud-based and physical data centers that use high-speed networks and support Remote Direct Memory Access (RDMA).

Trojan Attack. Most current neural Trojan attack methods involve inserting trojan during training rather than after the model has been deployed [115,116]. Rakin et al. [117] presented a novel technique known as Targeted Bit Trojan (TBT), which is an attack strategy involving flipping bits within a DNN model to insert a tailored neural trojan. Their method results in the production of a trigger that is uniquely crafted to locate the sensitive parts of the DNN weights held in the DRAM. They demonstrated that by flipping only a few vulnerable bits found by this method, a completely working DNN model may turn into a Trojan-infected model using existing bit-flip algorithms like row-hammer. The authors extensively evaluated CIFAR-10, SVHN, and ImageNet datasets on the VGG-16 and Resnet-18 algorithms. By adopting the ResNet-18 algorithm, TBT created an attack success rate of 91.93%. This was achieved by turning over only eighty-four bits out of eighty-eight million weight bits of the CIFAR-10 dataset.

Geometric Attack. Geometric attacks are relevant to visual objects like images or videos, essentially used in watermarking. This attack includes rotation, adding noise, translation, filtering, scaling, cropping, etc., of the object to cause trouble with the integrity of the IP. Tian [118] proposed a zero-watermarking method for videos to address these challenges and protect the IP without visual distortions. Those 3 aforementioned software IP attacks can be ranked as per the Table 4.

Fault-injection attacks can be operated in all sorts of software, starting from neural network models [71,119,120] to generic softwares [121]. Hence, defending a software IP against a fault-injection attack is critical and challenging. On the other hand, trojan attacks can have catastrophic effects on security-sensitive

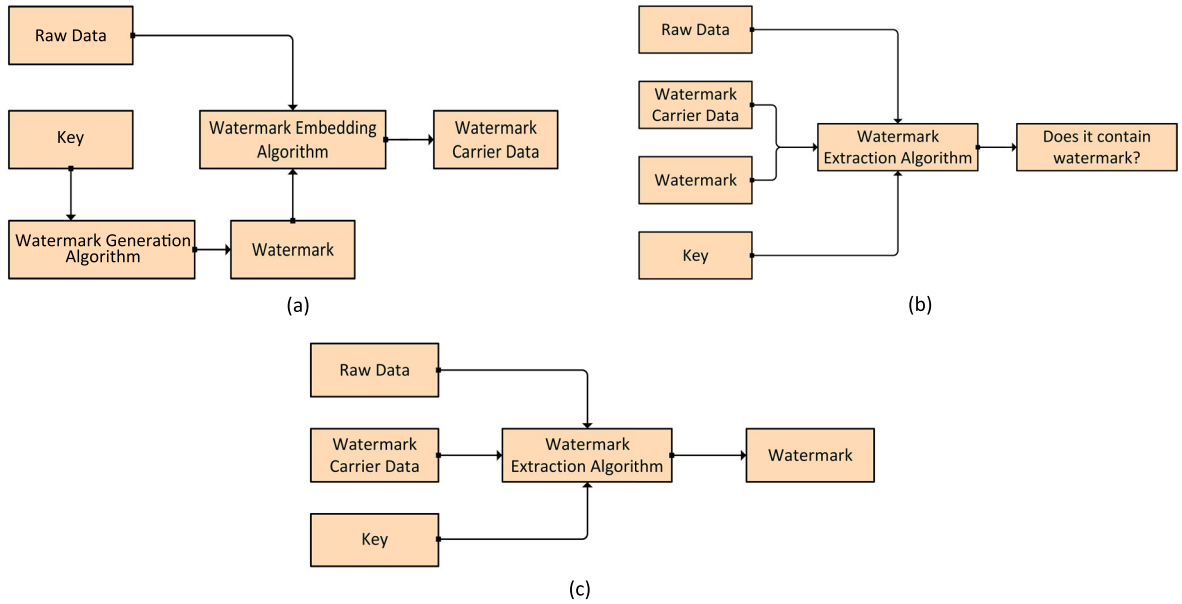

Fig. 5. Embedding, detection and extraction algorithms of watermarking. (a) Embedding watermark into the carrier data. (b) Detection of watermark within the carrier data. (c) Extraction of watermark from the carrier data.

programs. The covert nature of this attack and the numerous possibilities of its existence within the program make the detection process very difficult [74]. As a result, trojan attacks are similarly deleterious if not more than the fault-injection attacks. However, the geometric attacks are strongly connected to visual objects, which makes the attack limited to a specific IP, whereas the previous two attacks can be used on almost any kind of software IP.

3.2.2. Protection Schemes

Researchers have suggested various approaches to assure software IP protection, some of which have shown promising outcomes. Among the most notable of these strategies there are:

Fingerprinting and Watermarking. Watermarking is a technique that can protect models' IPs. Three aspects of watermarking (embedding, detection, and extraction) are illustrated in Figs. 5(a), 5(b), and 5(c), respectively. Kakikura et al. [34] suggested a white box watermarking approach based on modified Barni's approach for image watermarking. In this model, the watermark is integrated into the network parameters with no prior training, so this approach is suitable for pre-trained models. Furthermore, the suggested technique uses distinct keys to insert numerous watermarks within the neural networks.

A substantial amount of data and computing power are required for a DNN model to be appropriately trained from scratch. However, a pre-trained model takes less time to perform a task and often performs much better. So pre-trained DNN models should be protected since they are essential IPs. The secure fingerprinting DeepTrace framework is proposed in [35]. Using this model, owners can use a specific trigger set to discreetly fingerprint the target DNN model, which outputs will verify. This framework uses black-box and white-box validation, making it usable regardless of the known model specifics. The suggested DeepTrace framework can efficiently address the issue of fingerprinting DNN products.

Another software IP protection implemented in the cloud using deep learning is presented in [36], which outlines a model that enables owners to extract the fingerprint from a pirated model, implant a unique fingerprint for every client into the specifications of a DL model, and verify it. In the paper [37], the authors developed a watermark implanting technique to embed watermarks in DL models and develop a system for remote model ownership verification. In addition to preserving model fidelity for regular input data, their framework could promptly and reliably identify who owned each deep learning model deployed remotely.

Steganographic Technique. The steganographic technique can use five different media as the cover to hide the IP so that it remains intact from unauthorized exercises. These media are text, images, audio, video, and the network. Fig. 6(a) shows how messages are concealed on the sender's side of the channel during the generation stage. Fig. 6(b) shows an extraction on the receiver's side to uncover the message. This method is comparatively less famous because of the large overhead of the medium on which the IP is hidden. However, this method can be used on top of other methods to make the IP more secure from manipulation. Numerous propositions have been made regarding steganographic methods for protecting digital images [38–41]. In [42], Urbanovich et al. proposed a steganographic technique for software IP protection, which consists of color coordinate modifications in the text characters. A specific color model represents each character's (on-screen) color. The RGB model is used in the whole color spectrum of Microsoft Word 2007. The Least Significant Bits (LSB) of the color coordinates of an RGB spectrum are replaced by the binary representation of the plain text to create the cipher text. This paper concluded that if 4 bits are hidden in a single channel of RGB, users will not notice the changes if they do not know that steganography is being used.

Code Obfuscation. Zeng et al. [43] developed a Java byte code-based obfuscation system as an IP protection solution to deal with the challenges of reverse engineering. The experimental study revealed that the obfuscation technique described in this research not only increases the protection against IP infringements but also enhances the program's performance to some extent, allowing Java software's IP rights to be adequately protected.

Surprisal Analysis. Matrices are often used to compare how original assets are, especially when determining if IP rights (like design rights, copyright, etc.) are still valid. Sebastien Ragot, in his work, showed that using surprisal analysis and the principles of maximum entropy, the assets' originality can be stated as a function of distance (between the asset and its compounds) [44]. Unsupervised ML algorithms or any other distance computing

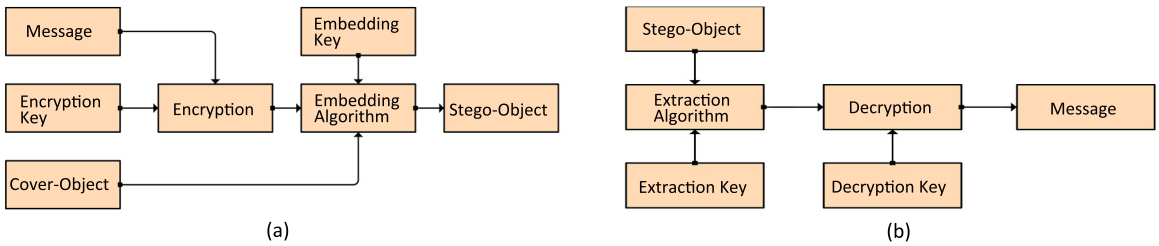

Fig. 6. Embedding and extraction algorithms of steganography. (a) Generation of stego-object in sender's side. (b) Extraction of original information in receiver's side.

Table 5
Common application domains based on IP types.

IP type	Application domain
Hardware IP	Field-Programmable Gate Array, Computational Forensic Engineering, Computer-Aided Design
Software IP	Deep Neural Network, Generative Adversarial Networks

algorithms can be used to determine the originality (like IP assets) based on the computed distance.

The present software IPP approaches place a significant emphasis on the information of legitimate end-users and the integrity of the whole computer system. Unfortunately, going through the previous works, we could not find a particular approach that can defend against every potential form of attack, and hence, based on the type of attack, different defense strategies are used. Unlike in hardware IPP, fingerprinting and watermarking play a crucial role in protecting software IPs effectively. Another comparatively newer concept explained above was steganography, which is robust against most attacks, especially man-in-the-middle attacks. Code obfuscation can also be promising, especially for a large code base. Surprisal analysis is the least used method of the above-described categories. Some other forms of defense approaches are crafted particularly for neural networks, following the technological advancement in AI and ML. Hence, software IPP schemes lean toward AI- and ML-based application domains and formulate attack-specific or domain-specific defenses.

4. Applications of IP protection technologies

IP protection is used in several applications and models both in terms of software and hardware of a system. The Table 5 presents a list of the application domains based on IP types. In addition to these two types there are a few miscellaneous applications which falls in the intersection of these two IPs.

4.1. Field-Programmable Gate Array

Since reusable digital designs are growing, concerns about intellectual design protection are rising to the center of attention. Lach et al. [23] first introduced a method to use Field Programmable Gate Array's (FPGA) unique properties to secure industrial IP investments. In this attempt, a watermark is applied to the structural configuration of the FPGA when a digital circuit is inserted into it. This watermark, which is difficult to locate, uniquely identifies the genesis of a circuit. Bit-streams in FPGA carry info about all hardware IP cores' functioning. If, in any way, the bit-stream can be accessed by an attacker, then he/she will be able to attack the IP. To resolve this crucial issue, an algorithm called WATERMARCH (a watermarking scheme) is proposed based on mutable architecture for IP protection using authenticated obfuscation [122]. This approach incorporates the watermark and the obfuscation into the initial design using a Hash-based Message Authentication Code (HMAC). Given the security of the underlying concealment and the fact that this technique incurs no additional costs, the watermark is practically "free".

The watermark can be recovered to show IP authorship or demonstrate the existence of malicious IP alteration. The FPGA architecture has no extra constraints, so this technology can be used for any model. WATERMARCH is a strong instrument for creating confidence in a dynamic sector for both IP owners and end users. To protect FPGA IPs from trespass, another watermarking method is proposed by Ganesh [123], which is publicly provable. In this method, the Data Matrix strategy and Zero-Knowledge proof are used. Time stepping is also used with the zero-knowledge proof protocol, which can defend against sensitive data leakage and cyber-attacks. Several scholars have proposed digital markings as a way to enable intellectual property ownership authentication. However, several of these methods have three flaws in common: *copy detection complexity, removal of a mark after it has been revealed for identity confirmation exposes a vulnerability, and lastly, due to incomplete mark removal, there are concerns with mark integrity*. Lach et al. [124] proposed a new method for watermarking FPGA IP using secure hash functions to counter these three flaws. It can create and embed several small watermarks, which are more verifiable, detectable, and secure than other IP safeguard methods.

4.2. Computational Forensic Engineering

Instead of depending on the content or design of watermarking, Wong et al. [125] proposed anx IP protection approach using Computational Forensic Engineering to determine which compiler or synthesis tool is used to create a specific design. Computational Forensic Engineering (CFE) aims to determine who generated a piece of IP. The generic CFE technique examines the characteristics of specific IP properties and quantifies the chance that they were produced from a well-known source. The proposed approach has four levels: data gathering of features and statistics, extraction of features, clustering of entities, and validation.

4.3. Computer-Aided Design

Inki et al. [126] introduced the first dynamic watermarking approach to protect Computer-Aided Design (CAD) IP value and compilation tool. This approach contains timing constraints and a collection of the author's signature encoding designs. The constraints are set to minimize hardware cost while encoding a signature that is difficult to track or delete. A generic method has been developed to hide data signatures in designs, and this can be used with any behavioral synthesis job (like assignment,

scheduling, transformations, etc.). In addition, error-correcting codes are employed to supplement the signature data's security against manipulation.

4.4. Deep Neural Network

Machine learning models are attractive targets of malevolent adversaries due to their high economic value and powerful learning abilities. For IP protection techniques related to DNN, Xue et al. [127] proposed six attributes of the taxonomy: scenario, capacity, mechanism, function, type, and target models. They also categorized different forms of attacks against DNN IPs into three tiers: model modifications, passive attacks, and active attacks. Depending on how easy it is to retrieve the DNN model's parameters, there are two primary types of attacks. The first is the white-box feature-based method, which considers having access to the model's parameters. The latter is a black-box trigger-set-based attack that considers the model's parameters are unknown [128]. In recent years, IP protection algorithms have used digital watermarking in DNN models during the learning phase, which is expected to be robust against model pruning, watermark overwriting, and model fine-tuning modifications [37, 129,130]. Another robust solution can be offered by multiple input–output training pairs, which can produce a surrogate model by learning just from its output. To implement that, a framework is proposed for model watermarking using a two-stage training approach to include watermarks inside the final model. One major flaw in this approach is that the hidden watermark can be discovered whenever the intruder trains a proxy model using the target model's input–output pairs [131].

Watermarking-based solutions cannot stop unauthorized or malicious users from using well-trained DNNs. Because of this, a novel framework is proposed using the Chaotic Map theory for protecting DNN providers' IP with little cost [132]. Instead of employing the traditional approach of encrypting the weight values, their method switches the weight locations to create a satisfying encryption effect while reducing data redundancy and decryption time. Furthermore, another novel method of watermarking for DNN is introduced using Deep Serial Number (DSN), for which unauthorized individuals will not be able to deploy the stolen model [133]. The implementation of the suggested DSN takes place within a knowledge-sharing framework, which involves first training a private teacher DNN, then distilling and transferring its knowledge to a succession of personalized student DNNs. Each client DNN is given a unique serial number throughout the distillation process. The integrated serial number might be used for ownership verification as a robust watermark. In [134], authors took this into consideration and proposed a novel DNN ownership verification method that relies on passport data. Their approach claimed to be resistant to ambiguity attacks and network alterations. The goal of incorporating digital passports is to build and train a DNN model that can distinguish the forged passports from the original ones. The DNN model inference findings are used in addition to the signatures on a passport to verify the authenticity.

Most of the above use cases do not consider neural networks from the perspective of distributed systems. Gomez et al. proposed a solution for protecting the IP of distributed neural networks, where trained DNNs are homomorphically encrypted while maintaining the privacy of interference and input data [135]. The proposed solution includes DNN modification using linear approximation and the deconstruction of every operation into multiplications and additions, and during the inference phase, the input data is encrypted. As a result of the revolutionary impact that deep learning has had on a wide range of industries and applications, such as language processing, speech synthesis, face identification, personalized ads, virtual assistants, fraud detection, generation of photo-realistic images, etc., it has become an absolute requirement to protect the existing DNN models from unauthorized duplication, dissemination, or exploitation [134].

4.5. Generative Adversarial Networks

Even though significant IP protection measures are available for Convolutional Neural Networks (CNN) and Deep Neural Network (DNN), Adversarial Generative Networks (GAN) are still entirely uninsured. To address image privacy, He et al. [62] proposed a GAN-based image privacy protection system called PriGAN. Each source image creates a private image using the Adversarial Image Perturbation (AIP) algorithm to deceive the recognition networks. Since the privacy image misleads the neural network, the original image remains intact. Using black-box and white-box contexts, the authors of [128] devised a verification mechanism to safeguard the intellectual property of GANs without sacrificing their performance. According to the authors, this method is resilient against ambiguity attacks, model pruning, and fine-tuning attacks.

4.6. Miscellaneous Applications

In the production and distribution of research, development, and educational resources, IP protection is a significant concern that we must address positively. IP protection is also used for safeguarding educational properties and data expressions with the help of machine learning based applications, real-time monitoring, and automation. Machine learning technologies can be used to create a data storage environment for educational resources that are highly integrated, subject-oriented, and constantly evolving. Machine learning's analysis-oriented and comprehensive decision-supporting mechanism can fully exploit the promising role of valuable discoveries and data integration and can also significantly protect complicatedly relevant and integrated resource data in education [136]. It can also protect E-commerce websites from being harmed by web crawlers and examine them to protect crucial website data from getting snatched [137]. However, because of the open nature of the testing platforms, it is impossible to provide an acceptable level of protection to the IP involved in crowd-sourced testing. In [138], Huang et al. offered an Interplanetary File System (IPFS) and a double encryption technique-based data access control strategy to address this issue for the IPs reliant on crowd-sourced models. Double encryption is achieved by combining an attribute-based encryption approach with a symmetric encryption process. Because of blockchains' non-tempering behavior traceability, the privacy safety of particular IP can be ensured during the crowd-sourced testing mode.

5. Challenges and future works

This study outlines the contemporary advancements in the hardware and software security of several IPs and their typical applications. Although significant progress has been made over the past two decades in IP protection research, significant research gaps still need to be addressed.

Hardware IP: The untrusted FEOL and BEOL foundries should be given more attention in split manufacturing. In particular, the answers to the questions "(i) can someone breaking into the FEOL foundry recognize the components missing from the BEOL foundry? (ii) can someone from the BEOL foundry recognize the missing components from the FEOL foundry?" need to be addressed conspicuously. In fingerprinting and watermarking techniques, it is still possible for a separate watermark to be

embedded by a third-party user to overwrite the initial watermark. Most currently used algorithms are yet to make substantial progress in resolving this problem. SLL is the most promising technology currently available when protecting IP infringements using logic locking. Nevertheless, sensitization attacks proved to be sufficient to break SLL, which made logic locking vulnerable to a known attack.

Software IP: In recent years, machine learning has become the most dominant topic in software IP protection research, but a few elements are connected to the ML-based approaches for dealing with software and hardware IP infringement. This includes scalability, transferability, data distribution dependency, noise resilience, transformation resilience, and adversarial technique resistance [134]. One particular machine learning model that has gained popularity in IP protection research is the DNN. DNN models are preferred due to their scalability, transferability, and data distribution dependencies. However, DNN does not excel at noise resilience, transformation resilience, or, most significantly, adversarial technique resistance. The vast majority of the currently available techniques to authenticate a DNN IP are passive authentication techniques, which can only check ownership rights after an infringement has already commenced. Active authentication techniques can resolve this issue but are tricky and challenging to implement. Also, more studies must be done to ensure data security within a DNN IP, despite data being an essential component of a DNN model. Most proposals utilize trivial datasets, such as IRIS, MNIST, CIFAR-10, etc., to simulate and evaluate, which are different from real-world datasets. Consequently, the results might be significantly different from the performance of these algorithms in practical situations. To have an unbiased evaluation of the algorithm, the models must be tested on real datasets that can prove the worth of the protection techniques.

System-level IP: In the end, even though hardware and software are inseparable components of a system, discussions regarding IP have traditionally been kept in two distinct categories. Therefore, combining the efforts of protecting software IP and hardware IP from various IP attacks and building a hybrid algorithm that can defend against attacks on system-level IP could be a possible area for further research in the future. In addition to hardware and software, data can be safeguarded with the differential privacy approach, which is a standard for evaluating the privacy concerns of a dataset [139,140].

6. Conclusion

Patents, copyrights, contracts, trademarks, trade secrets, etc. are all potential ways to safeguard intellectual property. However, the best strategy is determined by various criteria, including the type of IP to be protected, the technology's expected lifespan, the value of the IP, and its relevance to the individual or the organization. To tackle the threats of IP manipulation and theft, software-based and hardware-based measures must be taken apart from legal measures. This will ensure the total safety of the IP and make unethical people indifferent to IP manipulation and theft. For hardware-IP protection, hardware metering, hardware obfuscation, split manufacturing, IC camouflaging, logic locking, fingerprinting, and watermarking are the most prominent generic approaches. Over the last decade, FPGA and DNN, CNN, DL, GAN, etc. based IP protection techniques have been proposed, which fall under machine learning approaches. These techniques provide substantial immunity to the hardware-IPs. On the other hand, researchers have been seeking the methods that will be most effective in warding off software-IP threats. They have revisited digital watermarking, fingerprinting, steganographic techniques, code obfuscation, surprisal analysis, etc., and shown promising results against unlawful IP practices. Until now, research in this domain has been in its early stages, and much progress can be made, especially as ML is gradually expanding over the horizon of digital world. Hence, ML-based IPP that ensures both hardware and software-based IP protection concurrently under one umbrella might be the next big thing in IP protection research and development.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Z. Mingaleva, I. Mirskikh, Psychological aspects of intellectual property protection, Proc.-Soc. Behav. Sci. 190 (2015) 220–226.
[2] D.K. Sharma, Intellectual property and the need to protect it, Indian J. Sci. Res. 9 (1) (2014) 084–087.
[3] M. Delgado, M. Kyle, A.M. McGahan, Intellectual property protection and the geography of trade, J. Ind. Econ. 61 (3) (2013) 733–762.
[4] M. Rostami, F. Koushanfar, R. Karri, A primer on hardware security: Models, methods, and metrics, Proc. IEEE 102 (8) (2014) 1283–1295.
[5] J.A. Roy, F. Koushanfar, I.L. Markov, Ending piracy of integrated circuits, Computer 43 (10) (2010) 30–38.
[6] B. Colombier, L. Bossuet, Survey of hardware protection of design data for integrated circuits and intellectual properties, IET Comput. Digit. Tech. 8 (6) (2014) 274–287.
[7] R.S. Chakraborty, Hardware Security Through Design Obfuscation (Ph.D. thesis), Case Western Reserve University, 2010.
[8] X. Zhuang, T. Zhang, H.H.S. Lee, S. Pande, Hardware assisted control flow obfuscation for embedded processors, in: Proceedings of the 2004 International Conference on Compilers, Architecture, and Synthesis for Embedded Systems, 2004, pp. 292–302.
[9] D. Li, W. Liu, X. Zou, Z. Liu, Hardware IP protection through gate-level obfuscation, in: 2015 14th International Conference on Computer-Aided Design and Computer Graphics (CAD/Graphics), IEEE, 2015, pp. 186–193.
[10] A. Alaql, S. Chattopadhyay, P. Chakraborty, T. Hoque, S. Bhunia, LeGO: A learning-guided obfuscation framework for hardware IP protection, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. (2021).
[11] F. Koushanfar, G. Qu, Hardware metering, in: Proceedings of the 38th Annual Design Automation Conference, 2001, pp. 490–493.
[12] Y. Alkabani, F. Koushanfar, Active hardware metering for intellectual property protection and security, in: USENIX Security Symposium, 20, 2007, pp. 1–20.
[13] Y. Alkabani, F. Koushanfar, M. Potkonjak, Remote activation of ICs for piracy prevention and digital right management, in: 2007 IEEE/ACM International Conference on Computer-Aided Design, IEEE, 2007, pp. 674–677.
[14] J.A. Roy, F. Koushanfar, I.L. Markov, Protecting bus-based hardware IP by secret sharing, in: 2008 45th ACM/IEEE Design Automation Conference, IEEE, 2008, pp. 846–851.
[15] R.W. Jarvis, M.G. Mcintyre, Split manufacturing method for advanced semiconductor circuits, 2007, US Patent 7, 195, 931.
[16] K. Vaidyanathan, B.P. Das, E. Sumbul, R. Liu, L. Pileggi, Building trusted ICs using split fabrication, in: 2014 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), IEEE, 2014, pp. 1–6.
[17] K. Xiao, D. Forte, M.M. Tehranipoor, Efficient and secure split manufacturing via obfuscated built-in self-authentication, in: 2015 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), IEEE, 2015, pp. 14–19.
[18] J. Rajendran, O. Sinanoglu, R. Karri, Is split manufacturing secure? in: 2013 Design, Automation & Test in Europe Conference & Exhibition (DATE), IEEE, 2013, pp. 1259–1264.
[19] M. Li, B. Yu, Y. Lin, X. Xu, W. Li, D.Z. Pan, A practical split manufacturing framework for trojan prevention via simultaneous wire lifting and cell insertion, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 38 (9) (2018) 1585–1598.
[20] J. Rajendran, Y. Pino, O. Sinanoglu, R. Karri, Security analysis of logic obfuscation, in: Proceedings of the 49th Annual Design Automation Conference, 2012, pp. 83–89.
[21] J. Rajendran, H. Zhang, C. Zhang, G.S. Rose, Y. Pino, O. Sinanoglu, R. Karri, Fault analysis-based logic encryption, IEEE Trans. Comput. 64 (2) (2013) 410–424.
[22] J. Lach, W.H. Mangione-Smith, M. Potkonjak, Fingerprinting techniques for field-programmable gate array intellectual property protection, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 20 (10) (2001) 1253–1261.

[23] J. Lach, W.H. Mangione-Smith, M. Potkonjak, Signature hiding techniques for FPGA intellectual property protection, in: Proceedings of the 1998 IEEE/ACM International Conference on Computer-Aided Design, 1998, pp. 186–189.
[24] A.B. Kahng, J. Lach, W.H. Mangione-Smith, S. Mantik, I.L. Markov, M. Potkonjak, P. Tucker, H. Wang, G. Wolfe, Watermarking techniques for intellectual property protection, in: Proceedings of the 35th Annual Design Automation Conference, 1998, pp. 776–781.
[25] C.H. Chang, L. Zhang, A blind dynamic fingerprinting technique for sequential circuit intellectual property protection, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 33 (1) (2013) 76–89.
[26] L.W. Chow, J.P. Baukus, B.J. Wang, R.P. Cocchi, Camouflaging a standard cell based integrated circuit, 2012, US Patent 8, 151, 235.
[27] J. Rajendran, M. Sam, O. Sinanoglu, R. Karri, Security analysis of integrated circuit camouflaging, in: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, 2013, pp. 709–720.
[28] R.P. Cocchi, J.P. Baukus, L.W. Chow, B.J. Wang, Circuit camouflage integration for hardware IP protection, in: 2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC), IEEE, 2014, pp. 1–5.
[29] M.I.M. Collantes, M.E. Massad, S. Garg, Threshold-dependent camouflaged cells to secure circuits against reverse engineering attacks, in: 2016 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), IEEE, 2016, pp. 443–448.
[30] B. Erbagci, C. Erbagci, N.E.C. Akkaya, K. Mai, A secure camouflaged threshold voltage defined logic family, in: 2016 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), IEEE, 2016, pp. 229–235.
[31] I.R. Nirmala, D. Vontela, S. Ghosh, A. Iyengar, A novel threshold voltage defined switch for circuit camouflaging, in: 2016 21th IEEE European Test Symposium (ETS), IEEE, 2016, pp. 1–2.
[32] J. Davis, N. Kulkarni, J. Yang, A. Dengi, S. Vrudhula, Digital IP protection using threshold voltage control, in: 2016 17th International Symposium on Quality Electronic Design (ISQED), IEEE, 2016, pp. 344–349.
[33] B. Shakya, H. Shen, M. Tehranipoor, D. Forte, Covert gates: Protecting integrated circuits with undetectable camouflaging, IACR Trans. Cryptogr. Hardw. Embedd. Syst. (2019) 86–118.
[34] S. Kakikura, H. Kang, K. Iwamura, Collusion resistant watermarking for deep learning models protection, in: 2022 24th International Conference on Advanced Communication Technology (ICACT), IEEE, 2022, pp. 40–43.
[35] R. Wang, J. Kang, W. Yin, H. Wang, H. Sun, X. Chen, Z. Gao, S. Wang, J. Liu, DeepTrace: A secure fingerprinting framework for intellectual property protection of deep neural networks, in: 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), IEEE, 2021, pp. 188–195.
[36] G. Xu, H. Li, Y. Zhang, X. Lin, R.H. Deng, X. Shen, A deep learning framework supporting model ownership protection and traitor tracing, in: 2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS), IEEE, 2020, pp. 438–446.
[37] J. Zhang, Z. Gu, J. Jang, H. Wu, M.P. Stoecklin, H. Huang, I. Molloy, Protecting intellectual property of deep neural networks with watermarking, in: Proceedings of the 2018 on Asia Conference on Computer and Communications Security, 2018, pp. 159–172.
[38] B. Li, M. Wang, J. Huang, X. Li, A new cost function for spatial image steganography, in: 2014 IEEE International Conference on Image Processing (ICIP), IEEE, 2014, pp. 4206–4210.
[39] A. Tauhid, M. Tasnim, S.A. Noor, N. Faruqui, M.A. Yousuf, A secure image steganography using advanced encryption standard and discrete cosine transform, J. Inf. Secur. 10 (3) (2019) 117–129.
[40] K.A. Zhang, A. Cuesta-Infante, L. Xu, K. Veeramachaneni, SteganoGAN: High capacity image steganography with GANs, 2019, arXiv preprint arXiv:1901.03892.
[41] S.P. Lu, R. Wang, T. Zhong, P.L. Rosin, Large-capacity image steganography based on invertible neural networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 10816–10825.
[42] N. Urbanovich, V. Plaskovitsky, The use of steganographic techniques for protection of intellectual property rights, New Electr. Electron. Technol. Ind. Implement. (2011) 147–148.
[43] S. Zeng, X. Guo, Research on key technology of software intellectual property protection, in: 2021 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), IEEE, 2021, pp. 329–332.
[44] S. Ragot, Measuring the originality of intellectual property assets based on machine learning outputs, 2020, arXiv preprint arXiv:2010.06997.
[45] Y. Jin, D. Maliuk, Y. Makris, Post-deployment trust evaluation in wireless cryptographic ICs, in: 2012 Design, Automation & Test in Europe Conference & Exhibition (DATE), IEEE, 2012, pp. 965–970.
[46] T. Iwase, Y. Nozaki, M. Yoshikawa, 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.
[47] J. Li, L. Ni, J. Chen, E. Zhou, A novel hardware Trojan detection based on BP neural network, in: 2016 2nd IEEE International Conference on Computer and Communications (ICCC), IEEE, 2016, pp. 2790–2794.
[48] Y. Jin, 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.
[49] J. Balasch, B. Gierlichs, I. Verbauwhede, Electromagnetic circuit fingerprints for hardware trojan detection, in: 2015 IEEE International Symposium on Electromagnetic Compatibility (EMC), IEEE, 2015, pp. 246–251.
[50] N. Vashistha, H. Lu, Q. Shi, M.T. Rahman, H. Shen, D.L. Woodard, N. Asadizanjani, 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.
[51] M. Yan, B. Gopireddy, T. Shull, J. Torrellas, Secure hierarchy-aware cache replacement policy (SHARP): Defending against cache-based side channel attacks, in: 2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA), IEEE, 2017, pp. 347–360.
[52] H. Wang, H. Sayadi, S.M.P. Dinakarrao, A. Sasan, S. Rafatirad, H. Homayoun, Enabling micro AI for securing edge devices at hardware level, IEEE J. Emerg. Sel. Top. Circuits Syst. 11 (4) (2021) 803–815.
[53] K. Huang, J.M. Carulli, Y. Makris, Parametric counterfeit IC detection via support vector machines, in: 2012 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), IEEE, 2012, pp. 7–12.
[54] H. Dogan, D. Forte, M.M. Tehranipoor, Aging analysis for recycled FPGA detection, in: 2014 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), IEEE, 2014, pp. 171–176.
[55] U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, J. Schmidhuber, Modeling attacks on physical unclonable functions, in: Proceedings of the 17th ACM Conference on Computer and Communications Security, 2010, pp. 237–249.
[56] F. Kohnhäuser, A. Schaller, S. Katzenbeisser, PUF-based software protection for low-end embedded devices, in: International Conference on Trust and Trustworthy Computing, Springer, 2015, pp. 3–21.
[57] F. Ganji, S. Tajik, J.P. Seifert, Let me prove it to you: RO PUFs are provably learnable, in: ICISC 2015, Springer, 2015, pp. 345–358.
[58] F. Ganji, S. Tajik, F. Fäß ler, J.P. Seifert, Strong machine learning attack against PUFs with no mathematical model, in: International Conference on Cryptographic Hardware and Embedded Systems, Springer, 2016, pp. 391–411.
[59] X. Xu, J. Zhang, Rethinking FPGA security in the new era of artificial intelligence, in: 2020 21st International Symposium on Quality Electronic Design (ISQED), IEEE, 2020, pp. 46–51.
[60] Z. Xu, F. Yu, X. Chen, C. Liu, LanCeX: A versatile and lightweight defense method against condensed adversarial attacks in image and audio recognition, ACM Trans. Embedd. Comput. Syst. (TECS) (2022).
[61] A.S. Rakin, D. Fan, Defense-Net: Defend against a wide range of adversarial attacks through adversarial detector, in: 2019 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), IEEE, 2019, pp. 332–337.
[62] Y. He, C. Zhang, X. Zhu, Y. Ji, Generative adversarial network based image privacy protection algorithm, in: Tenth International Conference on Graphics and Image Processing (ICGIP 2018), Vol. 11069, SPIE, 2019, pp. 635–645.
[63] Y. Cao, D. Xu, X. Weng, Z. Mao, A. Anandkumar, C. Xiao, M. Pavone, Robust trajectory prediction against adversarial attacks, 2022, arXiv preprint arXiv:2208.00094.
[64] B.H. Hall, Patents and patent policy, Oxf. Rev. Econ. Policy 23 (4) (2007) 568–587.
[65] H.R. Varian, Copying and copyright, J. Econ. Perspect. 19 (2) (2005) 121–138.
[66] M. Harris, S.A. Ravid, S. Basuroy, Intellectual Property Contracts: Theory and Evidence From Screenplay Sales, Chicago Booth Research Paper (13–04), 2012.

[67] M. Grynberg, More than IP: Trademark among the consumer information laws, Wm. Mary L. Rev. 55 (2013) 1429.
[68] M.A. Lemley, The surprising virtues of treating trade secrets as IP rights, Law Theory Trade Secrecy (2011).
[69] P.J. Saidman, The glass slipper approach to protecting industrial designs or when the shoe fits, wear it, U. Balt. L. Rev. 19 (1989) 167.
[70] M. Potkonjak, A. Nahapetian, M. Nelson, T. Massey, Hardware Trojan Horse detection using gate-level characterization, in: 2009 46th ACM/IEEE Design Automation Conference, IEEE, 2009, pp. 688–693.
[71] A.S. Rakin, Z. He, D. Fan, Bit-flip attack: Crushing neural network with progressive bit search, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 1211–1220.
[72] T. Gu, K. Liu, B. Dolan-Gavitt, S. Garg, Badnets: Evaluating backdooring attacks on deep neural networks, IEEE Access 7 (2019) 47230–47244.
[73] S. Adee, The hunt for the kill switch, IEEE Spectr. 45 (5) (2008) 34–39.
[74] R.S. Chakraborty, S. Narasimhan, S. Bhunia, Hardware Trojan: Threats and emerging solutions, in: 2009 IEEE International High Level Design Validation and Test Workshop, IEEE, 2009, pp. 166–171.
[75] S. Bhunia, M.S. Hsiao, M. Banga, S. Narasimhan, Hardware Trojan attacks: Threat analysis and countermeasures, Proc. IEEE 102 (8) (2014) 1229–1247.
[76] K. Xiao, D. Forte, Y. Jin, R. Karri, S. Bhunia, M. Tehranipoor, Hardware Trojans: Lessons learned after one decade of research, ACM Trans. Des. Autom. Electron. Syst. (TODAES) 22 (1) (2016) 1–23.
[77] M.G. Rekoff, On reverse engineering, IEEE Trans. Syst. Man Cybern. (2) (1985) 244–252.
[78] U. Guin, K. Huang, D. DiMase, J.M. Carulli, M. Tehranipoor, Y. Makris, Counterfeit integrated circuits: A rising threat in the global semiconductor supply chain, Proc. IEEE 102 (8) (2014) 1207–1228.
[79] U. Guin, Q. Shi, D. Forte, M.M. Tehranipoor, FORTIS: A comprehensive solution for establishing forward trust for protecting IPs and ICs, ACM Trans. Des. Autom. Electron. Syst. (TODAES) 21 (4) (2016) 1–20.
[80] U. Guin, Z. Zhou, A. Singh, A novel design-for-security (DFS) architecture to prevent unauthorized IC overproduction, in: 2017 IEEE 35th VLSI Test Symposium (VTS), IEEE, 2017, pp. 1–6.
[81] N.G. Jayasankaran, A.S. Borbon, A. Abuellil, E. Sánchez-Sinencio, J. Hu, J. Rajendran, Breaking analog locking techniques via satisfiability modulo theories, in: 2019 IEEE International Test Conference (ITC), IEEE, 2019, pp. 1–10.
[82] M. Yasin, J.J. Rajendran, O. Sinanoglu, R. Karri, On improving the security of logic locking, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 35 (9) (2015) 1411–1424.
[83] P. Subramanyan, S. Ray, S. Malik, Evaluating the security of logic encryption algorithms, in: 2015 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), IEEE, 2015, pp. 137–143.
[84] L.W. Chow, W.M. Clark Jr., J.P. Baukus, Integrated circuit with reverse engineering protection, 2005, US Patent 6, 897, 535.
[85] J.P. Cocchi, B.J. Wang, L.W. Chow, P. Ouyang, Building block for a secure CMOS logic cell library, 2012, US Patent 8, 111, 089.
[86] M.E. Massad, S. Garg, M.V. Tripunitara, Integrated circuit (IC) decamouflaging: Reverse engineering camouflaged ICs within minutes., in: NDSS, 2015, pp. 1–14.
[87] D. Liu, C. Yu, X. Zhang, D. Holcomb, Oracle-guided incremental SAT solving to reverse engineer camouflaged logic circuits, in: 2016 Design, Automation & Test in Europe Conference & Exhibition (DATE), IEEE, 2016, pp. 433–438.
[88] Z. Huang, Q. Wang, Y. Chen, X. Jiang, A survey on machine learning against hardware trojan attacks: Recent advances and challenges, IEEE Access 8 (2020) 10796–10826.
[89] K.G. Liakos, G.K. Georgakilas, S. Moustakidis, N. Sklavos, F.C. Plessas, Conventional and machine learning approaches as countermeasures against hardware trojan attacks, Microprocess. Microsyst. 79 (2020) 103295.
[90] M. Fyrbiak, S. Strauß, C. Kison, S. Wallat, M. Elson, N. Rummel, C. Paar, Hardware reverse engineering: Overview and open challenges, in: 2017 IEEE 2nd International Verification and Security Workshop (IVSW), IEEE, 2017, pp. 88–94.
[91] S. Potluri, A. Aysu, A. Kumar, Seql: Secure scan-locking for IP protection, in: 2020 21st International Symposium on Quality Electronic Design (ISQED), IEEE, 2020, pp. 7–13.
[92] M. Yasin, B. Mazumdar, J.J.V. Rajendran, O. Sinanoglu, SARLock: SAT attack resistant logic locking, in: 2016 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), IEEE, 2016, pp. 236–241.
[93] Y. Xie, A. Srivastava, Mitigating SAT attack on logic locking, in: International Conference on Cryptographic Hardware and Embedded Systems, Springer, 2016, pp. 127–146.
[94] A. Chakraborty, A. Mondai, A. Srivastava, Hardware-assisted intellectual property protection of deep learning models, in: 2020 57th ACM/IEEE Design Automation Conference (DAC), IEEE, 2020, pp. 1–6.
[95] B.F. Goldstein, V.C. Patil, V.C. Ferreira, A.S. Nery, F.M.G. França, S. Kundu, Preventing DNN model IP theft via hardware obfuscation, IEEE J. Emerg. Sel. Top. Circuits Syst. 11 (2) (2021) 267–277.
[96] S. Patnaik, M. Ashraf, H. Li, J. Knechtel, O. Sinanoglu, Concerted wire lifting: Enabling secure and cost-effective split manufacturing, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 41 (2) (2021) 266–280.
[97] M. Yasin, B. Mazumdar, O. Sinanoglu, J. Rajendran, CamoPerturb: Secure IC camouflaging for minterm protection, in: 2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), IEEE, 2016, pp. 1–8.
[98] J.A. Roy, F. Koushanfar, I.L. Markov, EPIC: Ending piracy of integrated circuits, in: Proceedings of the Conference on Design, Automation and Test in Europe, 2008, pp. 1069–1074.
[99] Y. Lee, N.A. Touba, Improving logic obfuscation via logic cone analysis, in: 2015 16th Latin-American Test Symposium (LATS), IEEE, 2015, pp. 1–6.
[100] S.M. Plaza, I.L. Markov, Solving the third-shift problem in IC piracy with test-aware logic locking, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 34 (6) (2015) 961–971.
[101] N. Asadizanjani, M. Tehranipoor, D. Forte, Counterfeit electronics detection using image processing and machine learning, in: Journal of Physics: Conference Series, Vol. 787, IOP Publishing, 2017, 012023.
[102] K. Huang, Y. Liu, N. Korolija, J.M. Carulli, Y. Makris, Recycled IC detection based on statistical methods, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 34 (6) (2015) 947–960.
[103] C. Bao, D. Forte, A. Srivastava, On application of one-class SVM to reverse engineering-based hardware Trojan detection, in: Fifteenth International Symposium on Quality Electronic Design, IEEE, 2014, pp. 47–54.
[104] A. Kulkarni, Y. Pino, T. Mohsenin, SVM-based real-time hardware Trojan detection for many-core platform, in: 2016 17th International Symposium on Quality Electronic Design (ISQED), IEEE, 2016, pp. 362–367.
[105] K. Hasegawa, M. Oya, M. Yanagisawa, 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.
[106] C. Bao, D. Forte, A. Srivastava, On reverse engineering-based hardware Trojan detection, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 35 (1) (2015) 49–57.
[107] K. Hasegawa, M. Yanagisawa, N. Togawa, Trojan-feature extraction at gate-level netlists and its application to hardware-Trojan detection using random forest classifier, in: 2017 IEEE International Symposium on Circuits and Systems (ISCAS), IEEE, 2017, pp. 1–4.
[108] B. Singh, D. Evtyushkin, J. Elwell, R. Riley, I. Cervesato, On the detection of kernel-level rootkits using hardware performance counters, in: Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, 2017, pp. 483–493.
[109] T. Liu, W. Wen, Y. Jin, SIN 2: Stealth infection on neural network—a low-cost agile neural trojan attack methodology, in: 2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), IEEE, 2018, pp. 227–230.
[110] M. Alam, S. Bhattacharya, S. Dutta, S. Sinha, D. Mukhopadhyay, A. Chattopadhyay, RATAFIA: ransomware analysis using time and frequency informed autoencoders, in: 2019 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), IEEE, 2019, pp. 218–227.
[111] Y. Kim, R. Daly, J. Kim, C. Fallin, J.H. Lee, D. Lee, C. Wilkerson, K. Lai, O. Mutlu, Flipping bits in memory without accessing them: An experimental study of DRAM disturbance errors, ACM SIGARCH Comput. Archit. News 42 (3) (2014) 361–372.
[112] J. Li, A.S. Rakin, Y. Xiong, L. Chang, Z. He, D. Fan, C. Chakrabarti, Defending bit-flip attack through DNN weight reconstruction, in: 2020 57th ACM/IEEE Design Automation Conference (DAC), IEEE, 2020, pp. 1–6.
[113] M. Javaheripi, J.W. Chang, F. Koushanfar, AccHashtag: Accelerated hashing for detecting fault-injection attacks on embedded neural networks, ACM J. Emerg. Technol. Comput. Syst. (JETC) (2022).
[114] A. Tatar, R.K. Konoth, E. Athanasopoulos, C. Giuffrida, H. Bos, K. Razavi, Throwhammer: Rowhammer attacks over the network and defenses, in: 2018 USENIX Annual Technical Conference (USENIX ATC 18), 2018, pp. 213–226.
[115] T. Gu, B. Dolan-Gavitt, S. Garg, Badnets: Identifying vulnerabilities in the machine learning model supply chain, 2017, arXiv preprint arXiv:1708.06733.
[116] Y. Gao, C. Xu, D. Wang, S. Chen, D.C. Ranasinghe, S. Nepal, Strip: A defence against Trojan attacks on deep neural networks, in: Proceedings of the 35th Annual Computer Security Applications Conference, 2019, pp. 113–125.
[117] A.S. Rakin, Z. He, D. Fan, Tbt: Targeted neural network attack with bit Trojan, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 13198–13207.
[118] L. Tian, A zero-watermarking method to protect intellectual property under strong geometric attacks, in: 2017 2nd International Conference on Multimedia and Image Processing (ICMIP), IEEE, 2017, pp. 172–176.

[119] Y. Liu, L. Wei, B. Luo, Q. Xu, Fault injection attack on deep neural network, in: 2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), IEEE, 2017, pp. 131–138.
[120] F. Yao, A.S. Rakin, D. Fan, {*DeepHammer*}: Depleting the intelligence of deep neural networks through targeted chain of bit flips, in: 29th USENIX Security Symposium (USENIX Security 20), 2020, pp. 1463–1480.
[121] F. Brasser, L. Davi, D. Gens, C. Liebchen, A. Sadeghi, Can't touch this: Practical and generic software-only defenses against rowhammer attacks, 2016, arXiv preprint arXiv:1611.08396.
[122] B. Olney, R. Karam, WATERMARCH: IP protection through authenticated obfuscation in FPGA bitstreams, IEEE Embedd. Syst. Lett. 13 (3) (2020) 81–84.
[123] S.G. Ramasamy, Watermark decoding technique using machine learning for intellectual property protection, Int. J. New Pract. Manage. Eng. 8 (03) (2019) 01–09.
[124] J. Lach, W.H. Mangione-Smith, M. Potkonjak, Robust FPGA intellectual property protection through multiple small watermarks, in: Proceedings 1999 Design Automation Conference (Cat. No. 99CH36361), IEEE, 1999, pp. 831–836.
[125] J.L. Wong, D. Kirovski, M. Potkonjak, Computational forensic techniques for intellectual property protection, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 23 (6) (2004) 987–994.
[126] I. Hong, M. Potkonjak, Behavioral synthesis techniques for intellectual property protection, in: Proceedings 1999 Design Automation Conference (Cat. No. 99CH36361), IEEE, 1999, pp. 849–854.
[127] M. Xue, J. Wang, W. Liu, DNN intellectual property protection: Taxonomy, attacks and evaluations, in: Proceedings of the 2021 on Great Lakes Symposium on VLSI, 2021, pp. 455–460.
[128] D.S. Ong, C.S. Chan, K.W. Ng, L. Fan, Q. Yang, Protecting intellectual property of generative adversarial networks from ambiguity attacks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 3630–3639.
[129] Y. Adi, C. Baum, M. Cisse, B. Pinkas, J. Keshet, Turning your weakness into a strength: Watermarking deep neural networks by backdooring, in: 27th {*USENIX*} Security Symposium ({*USENIX*} Security 18), 2018, pp. 1615–1631.
[130] H. Chen, B.D. Rohani, F. Koushanfar, Deepmarks: A digital fingerprinting framework for deep neural networks, 2018, arXiv preprint arXiv:1804.03648.
[131] J. Zhang, D. Chen, J. Liao, W. Zhang, H. Feng, G. Hua, N. Yu, Deep model intellectual property protection via deep watermarking, IEEE Trans. Pattern Anal. Mach. Intell. (2021).
[132] N. Lin, X. Chen, H. Lu, X. Li, Chaotic weights: A novel approach to protect intellectual property of deep neural networks, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 40 (7) (2020) 1327–1339.
[133] R. Tang, M. Du, X. Hu, Deep serial number: Computational watermarking for DNN intellectual property protection, 2020, arXiv preprint arXiv:2011.08960.
[134] L. Fan, K.W. Ng, C.S. Chan, Q. Yang, DeepIP: Deep neural network intellectual property protection with passports, IEEE Trans. Pattern Anal. Mach. Intell. (2021).
[135] L. Gomez, A. Ibarrondo, J. Márquez, P. Duverger, Intellectual property protection for distributed neural networks-towards confidentiality of data, model, and inference, in: ICETE (2), 2018, pp. 313–320.
[136] J. Cao, Mode optimization and rule management of intellectual property rights protection of educational resource data based on machine learning algorithm, Complexity 2021 (2021).
[137] M. Mei, H. Tan, Data expression and protection of intellectual property education resources based on machine learning, Complexity 2021 (2021).
[138] S. Huang, Z. Yang, C. Zheng, J. Wan, An intellectual property data access control method for crowdsourced testing system, in: 2021 8th International Conference on Dependable Systems and their Applications (DSA), IEEE, 2021, pp. 434–438.
[139] H. Jiang, J. Pei, D. Yu, J. Yu, B. Gong, X. Cheng, Applications of differential privacy in social network analysis: A survey, IEEE Trans. Knowl. Data Eng. 35 (1) (2021) 108–127.
[140] H. Jiang, Y. Gao, S. Sarwar, L. GarzaPerez, M. Robin, Differential privacy in privacy-preserving big data and learning: Challenge and opportunity, in: Silicon Valley Cybersecurity Conference: Second Conference, SVCC 2021, San Jose, CA, USA, December 2–3, 2021, Revised Selected Papers, Springer, 2022, pp. 33–44.