Abstract:

As the Internet of Things (IoT) technology advances, billions of multidisciplinary smart devices act in concert, rarely requiring human intervention, posing significant challenges in supporting trusted computing and user privacy, as well as protecting against attacks such as spoofing, denial of service (DoS), jamming, and eavesdropping. To tackle attacks on the IoT and cyber-physical ecosystem, many intrusion detection and security approaches have been presented in the literature. Machine learning (ML) based intrusion and anomaly detection has lately gained traction due to its capacity to cope with encrypted and rapidly developing threat techniques. This work investigates into machine learning (ML) and deep learning (DL) methodologies for IoT device security and examine the benefits, drawbacks, and potential. To protect an IoT infrastructure, various solutions look into hardware-based methods for ML-based IoT authentication, access control, secure offloading, and malware detection schemes. This review aims to illuminate the value of various approaches for addressing IoT security in a truly effective, flexible, and seamless manner, as well as to provide answers to questions about tradeoffs in integrating accelerators and customizing embedded device architectures for effective use of ML-based methods.

Surveying into machine learning (ML) and deep learning (DL) methodologies for IoT device security while investigating the benefits, drawbacks, and potential of blending h...View more

Published in: IEEE Access (Volume: 10)

Page(s): 58603 - 58622

Date of Publication: 30 May 2022

Electronic ISSN: 2169-3536

DOI: 10.1109/ACCESS.2022.3179047

Publisher: IEEE

Funding Agency:

SECTION I.

Introduction

To ensure end-to-end secure cyberphysical systems with Internet-of-Things (IoT) infrastructures, one important parameter ignored till recently, necessitates the integration of a substantial level of cyber protection of IoT end devices, namely, of the microcontroller units (MCUs) and their interconnection networks [1]–[5]. To tackle software and hardware threats, modern embedded Systems-on-Chip (SoCs) are designed considering compliance with the requirements of the ARM Trusted Base System architecture [6], i.e., base SoC isolation, cryptography trusted boot, and debug protection. ARM TrustZone hardware organization is used to isolate the target application from other applications at runtime by providing a partitioning of internal and

external memory into trusted and non-trusted worlds. By leveraging this principle, numerous typical security features for IoT connected devices are fulfilled, such as secure boot, secure firmware installation, cryptographic accelerators, secure data storage and secure firmware update [7]. Additional protection of applications and data on such devices includes several techniques ranging from software to specialized circuitry, such as emerging instruction set extensions, ARM's Branch Target Identification (BTI) and Memory Tagging Extension (MTE) to mitigate memory-related security bugs [8], [9]. It is equally of growing concern to ensure built-in active tamper detection, by reducing the vulnerability surface of encryption of symmetric accelerators (e.g., AES) and asymmetric public-key accelerators (PKA) against attacks with side-channel analysis (SCA), secure hardware and independent keys for persistent data storage. Additionally, security countermeasures may also support internal monitoring of perturbation attacks to erase secret data, according to PCI security standards council requirements for end-point applications and data [10]. Attestation along with Hardware Security Managers (HSM) and Physically Unclonable Functions (PUF) have also been proposed for integrity protection [11]–[13]. HSM units are employed to orchestrate not only the key distribution, but also cryptography related operations such as authentication, cryptography-based trusted boot and debug protection. Moreover, security strategies extend beyond hardware design protection (e.g., against the insertion of hardware Trojans during the production phase, through netlist obfuscation provided by logic locking) through firmware and operating system layers, such as various secure, trusted, and verified microkernel architectures [14].

Nonetheless, IoT devices are mostly restricted devices with inadequate tamper-resistant and tamper-detection methods, allowing connected devices to leak personal data, for example, by allowing modified firmware to access authentication credentials. With the rapid proliferation of Internet of Things (IoT) devices and cloud systems, most cyber-physical systems' quickly increasing attack surface can scarcely keep attacks from multiplying in quantity and sophistication, as seen in Fig 1. IoT-enabled cyber-physical systems (CPSs) in factories, smart grids, and automobiles now have a variety of communication interfaces, remote monitoring, and software-over-the-air administration capabilities [2], [15]. In essence, Industrial IoT (IIoT) networks have a broad attack surface, making a covert channel more difficult to detect. As a result, most intrusion detection and prevention systems (IDS/IPS) rely on signature originality or signatures that have not been tampered with. Meanwhile, traditional IT network isolation is no longer feasible for IIoT networks. This is due to cautious malware design with obfuscation, or attempts to probe or otherwise manipulate devices and network, can easily bypass signature-based IDS which use matching patterns. Smarter and more advanced boundary control and auditing of access in needed against the trust boundaries of IIoT networks [16]. To address such challenges, artificial intelligence (AI) is promising effective solutions pertaining to security.

FIGURE 1.

Threat model involving a wide spectrum of attacks spreading mainly in IoT authentication, access control, secure offloading, and malware detection, in IoT infrastructures facilitating integration between the physical world and computer communication networks, and applications (apps) [17]–[19].

Show All

Essentially, security becomes increasingly complex as the attack surface of computing things increases. Because of the restricted things resources available, key concepts and common security mechanisms may need to be shared throughout layers of security solutions for each one [20]–[23]. Even the integration of micro-architectural features in the latest processors may extend the scope for new side-channel attacks. Performance counters, for example, can indicate branch misses events to aid successful attacks on asymmetric ciphers like RSA, as demonstrated recently [24]. Meanwhile, hostile attackers are becoming more sophisticated, frequent, and automated, even using AI-based approaches to automate IoT security breaches and enable more successful but also less detectable attacks [25], [26]. For instance, recent developments reveal machine learning framework for side-channel attacks on asymmetric cryptography, such as RSA and ECC, that analyzes leakage in multiple side-channel traces, identifying the best trace for key retrieval on a 32-bit ARM Cortex-M4 microcontroller [27].

A. Machine Learning for IoT Security

Fundamentally, using machine learning methodologies involves a threefold scope, (i) to facilitate an effective attack against an IoT infrastructure by exploiting a hardware, software or network vulnerability; (ii) to establish a robust and automated detection and protection system against malware, side-channel threats, fault attacks, and other threats; and (iii) to create the need to implement countermeasures against adversaries to the ML-based techniques themselves.

Machine learning and deep learning algorithms are rapidly being used in cybersecurity applications such as intrusion and virus detection, user authentication (e.g., biometrics), and user privacy. These advanced learning methods may be used to evaluate and learn from underlying IoT data in order to enhance threat assessment and attack detection, and thereby identify breaches in the IoT ecosystem. Deep learning approaches that adapt and evolve at the same time can not avoid sophisticated threats like entity or object profiling, as well as possible interdependent vulnerabilities and exploits. Deep learning can significantly change the cybersecurity landscape. For example, to improve traditional techniques that use pattern-matching to detect malware, such as by using register values and states to identify original identity of industrial embedded devices [28]. These pattern matching solutions can barely match the increasing rate of new attacks and variants. Sophisticated malware has been able to bypass or infiltrate network and end-point detection strategies, thus continously sporting significant cyber-attacks. Additionally, a huge number of IoT devices are lacking in processing power and storage capacity to run security solutions and maintain databases of threat and malware signatures to protect them against threats. On top, even detection methods based on observing anomalies present weaknesses, since activities which users rarely perform may also be classified as an anomaly [29]. In this scope, deep learning can be leveraged to learn and evolve new defense mechanisms using all available data and address the growing cybersecurity challenge [30], [31].

Essentially, pertaining to security of IoT devices and networks, the emerging ML and DNN techniques and branches (e.g., reinforcement learning (RL), Long short term memory (LSTM), generative adversarial network (GAN)) bring the following benefits.

- Machine learning methods help in automated threat detection and prevention by addressing complexity of modeling an indefinite space of malicious behaviours, and by integrating analysis, detection and protection systems.
- Machine learning methods can manage huge number of devices, to navigate their firmware updates and security patches; AI-driven policy and update management can help for firmware updates and patches to apply in all devices in a timely fashion.
- Machine learning for cybersecurity is scalable-independent as it makes it possible for a system to learn by its own experience as it grows and self-tune to become increasingly efficient and effective.

These advantages, especially, are more valuable in view of combining ML and DNN methods with the increased difficulty to tamper hardware-based techniques. As the security of modern embedded computing devices raises extensive concerns, hardware-based monitors and countermeasures offer increased guarantees when developed and deployed to thwart various cyber attacks. Moreover, hardware-based detection techniques require smaller overhead for resource and latency compared to the software-based counterparts. Several such techniques heavily utilize machine learning (ML) techniques [32]–[34], thus attempting to raise a strong defense umbrella against the numerous threats and attacks. However, the landscape of securing IoT environments using ML methods includes numerous challenges, stemming from the subtle attributes of the various attacks,[1] combined with software and hardware circuitry complexity with security-weak surfaces. This marriage of ML algorithms with secure- and trusted- conscious methods, spanning hardware and software layers, is proving to grow as a two-edged sword in IoT environments [35], as analyzed next in this article.

B. Related Surveys

Prior works provide surveys that deliver insight into several related topics, without delivering though a unified, comprehensive view on modern research efforts in ML and DNN methods combined with microarchitectural techniques for secure and trusted edge computing. Table 1 summarizes distinctive surveys on secure architectures for trusted computations, on advancements on machine learning practices in IoT and on the intersection between intrusion detection, between hardware acceleration methods of machine learning and on emerging IoT infrastructures.

TABLE 1 Background Research Contributions Presenting Surveys, Analyses, Taxonomies and Future Perspectives

To the best of my knowledge, this investigation work is a systematic comprehensive review that analyzes different strategies and presents the effectiveness and practical perspectives of machine learning powered methods assisted by hardware techniques and accelerators for the security of IoT devices and systems. In particular, we discuss these techniques, their merits and drawbacks, summarize strengths and weaknesses in hardware-based ML domain for intrusion detection research and suggest future research challenges. The aim of this work is, first, to showcase if the the gap between the capabilities of machine learning (ML) and deep learning (DL) and the requirements of the IoT resource-constrainted environment can be effectively

bridged. This analysis is balanced against today's and emerging cutting-edge microarchitectural advancements, with a view towards addressing the security challenges of the IoT ecosystems. Figure 2 shows the concepts, dimensions investigated in this article, with particular focus on prominent hardware solutions that leverage ML methods towards securing IoT devices.

FIGURE 2.

Security objectives and ML methods explored in the scope of enabling efficient use of hardware and software techniques for secure IoT infrastructures.

Show All

The rest of the paper is organized as follows. Section II discusses anomaly detection for IoT devices and ecosystem. Section III surveys the literature ML-based hardware methods for security investigating the marriage between ML for systems protection and especially forensics for IoT systems. Section IV reviews and analyzes research and industrial techniques for enhancing IoT security at the edge while bringing ML and DL in support as well. Then, section V provides insight on the effectiveness of the various trends and techniques and identify research gaps that deserve further research efforts. Section VI presents conclusions and suggests future research directions.

SECTION II.

Anomaly Detection in IoT Embedded Systems, Challenges and Opportunities

Anomaly detection methods are developed to mitigate various threats, such as false data injection attacks, denial of service, or compromised firmware, in different Cyber-Physical Systems (CPSs) domains. This is increasingly important in industrial IoT infrastructures, due to serious and wide impact as shown in Figure 3; especially today, with modern IoT multi-core integrated devices that provide rich functionalities but also wide attack surface, including device, network and cloud. Intrusion detection methods are mostly dedicated to ensuring network communication security [39], but these methods are undermined from IoT device heterogeneity and the highly dynamic threat landscape against them.

FIGURE 3.

Impact dimensions of threat model to IoT infrastructures.

Show All

Machine learning approaches do not rely on domain-specific knowledge, but they usually require a large quantity of labeled data through, for instance, classification-based methods [50]. An inherent requirement to guarantee tamper-resistant CPSs involves specification of accurate adversary models, that is, (i) a complete specification of all known attack vectors including risk

assessment of identified attacks (i.e., the likelihood of the attack and the impact of exploiting each threat on the system) and, (ii) maintaining of this attack database current. On the contrary, the key advantage of behavior-based approaches based on unsupervised techniques is that they do not focus on something specific. Although these approaches can be susceptible to false positives, they are independent from any past knowledge of attack methods and their impact. However, current anomaly-based detection systems can hardly detect new types of attacks, because they are designed either for specific applications, or for limited environments. Thus, the defense capability of existing security mechanisms can be mediocre, and for instance, limited to specific distributed denial-of-service (DDoS) attacks. As these attacks can spitefully diversify the underlying protocol or the operation method, the fundamental features of DDoS attacks should become the basis of any detection method. Several research works focus on detecting the attacks by using machine learning techniques [17], [40] and in IoT infrastructures [18]. Nonetheless, it is challenging to match the most successful detection technique to the attributes of the attack surface of IoT infrastructures in a hollistic way, or in a specific-optimized way. In this context, researchers have showed how to use even twenty-three features to detect DDoS attacks using various ML classifiers [51].

In addition to cyber-physical attacks, as it is not easy to distinguish the cause of such an abnormal situation in a given system, either a fault or an attack, detection and prevention techniques should consider both interchangeably. Faults are an abnormal state which might lead to errors or failure of the system, including permanent, transient, or intermittent, raising important concerns and defenses in industry [52], automotive [53] and medical [54] domains. Physical defects in the sensors, inside the chip, or concurrent attacks to the IoT device can be a major cause of damage to the system.

IoT infrastructures are exploding in industry, automotive, healthcare, while integrating different networks and different devices which makes it a nearly unreachable target to learn anomalous data that cause physical damage in view of unknown attack vectors and attack techniques. Due to the lack of anomalous data collections for training, ML-based detectors can hardly provide high accuracy with an adverse effect involving false alarms. Thus, there are limits to the ability to generate anomalous data such as car accident data and Cyber-physical System (CPS) faults in medicine.

Challenges for embedded devices in IoT infrastructures mainly involve spatial and temporal relationships, devices and data heterogeneity and labeled data shortage. Time-series data generation, transmission (and even prediction [55], or predicting of the future timestamp while avoiding a range of anomalies such as point anomalies, contextual anomalies, and discords in time series data [56]) are widely accepted in several domains. In this scope, recurrent neural networks (RNN) have been investigated and shown superior performance for behavior modeling. Among them, the long short-term memory (LSTM) emerged as an enhanced version of RNN for deep anomaly detection framework for sensing time-series data in Industrial IoT (IIoT) [35], while some also using multi-dimensional sensor data fusion [57]. Several intrusion and attack recognition works [35], [58] have demonstrated the efficiency of the RNN in terms of discovering anomalies in an accurate and timely way. Even though convolution neural networks (CNNs) are inpractical to capture sequential data representations, they have been leveraged for intrusion detection due to their capability to extract spatial features. By including parallelism techniques, temporal CNN has been introduced and validated to be more efficient over the CNN since temporal convolutional network (TCN) can learn windowed temporal dependencies over long spans more convincingly [59]. Additionally, TCN exhibits improved performance

compared to LSTM in many sequence problems, while at the same time, RNNs design is more complex compared to TCN. Unsupervised and supervised learning (i.e., deep learning with random forest) can be combined in even more sophisticated ways to provide the ability to adapt to assault patterns with rapid changes. With the extremely large trust boundaries of IIoT networks, such joined strategy can enable automatic updating of the detection engine knowledge base, thus, protecting trust boundaries of IIoT networks from zero day attacks [60].

In principle, to defend the network, all of the ML-based approaches listed above work at the host level. However, malicious programs operating on IoT devices, as well as wireless attacks, necessitate edge security countermeasures that are strengthened with machine learning approaches. Furthermore, because the trust boundary incorporates all of these elements, solutions at the edge level should be harmonized with typical conventional protection techniques that are already in use on cloud and SCADA servers, as well as databases. These conventional protection schemes may include monitoring and logging systems, remote access and anonymization control, and smart configuration and changes management.

SECTION III.

Hardware-Assisted ML for Security

Essentially, an IoT device is mandatory to be secured through a chain of trust. This chain is developed when an IoT device boots up only if cryptographically signed code components are first executed. These software components include bootloaders, kernel and kernel extensions, all the way from bootloader to userspace. For signing the software components, a trusted entity is responsible to provide signatures by using public-key cryptography. In particular, to establish secure boot of the microcontroller, an authentic first piece of software should be locked in a flash memory region sealed from further programming and, should implement the digital signature check of the next piece of software. In addition, the keys should be stored in specialized secure hardware to prevent not only modification but also indirect or partial extraction. The ultimate objective is to ensure a root of trust, essentially meaning that the embedded system is unclonable. Trusted paths, channels and secure communication connections (e.g., via differentiated keys, two-way communication, chained certificates) between secure signing entity, trusted module and firmware components are built on the base of authentication keys and their certificates. A secure embedded system needs to satisfy all the security requirements that involve the authenticity of the running software, the confidentiality of permanently stored elements (keys and sensitive data), and run-time state integrity.

All components of the system (i.e., components in the hardware architecture), as well as executing software and networking data, should be analyzed to find aberrant behavior that suggests security violations in a computing system. Without incorporating machine-learning algorithms, the analysis and identification activities in this process pose significant obstacles. A Network-based Intrusion Detection System (NIDS) can monitor traffic and analyze packets, hosts, and service flows to look for possible attacks in this context. This procedure is divided into two parts: algorithms for implementing effective inference techniques, such as machine learning, and model building for generating an attack profile and classification for determining if the examined traffic is valid or vulnerable to attack.

The following sections aim to illustrate how current research has tackled speeding up machine learning algorithms to make them suitable for IoT devices, as well as whether these specific

acceleration approaches may be used for security provisioning. Figure 4 shows the blending of domains investigated in the remaining sections of this work.

FIGURE 4.

Surveying of security and machine learning research methods blending hardware and software techniques in IoT infrastructures.

Show All

A. ML and DNN Accelerators

Most research works in ML and DNN accelerators focus on computer vision domain, biomedical signal processing, etc. However, these works pave the way for integrations of such accelerators also in securing IoT infrastructures, devices and networks from anomalies.

Hardware accelerators typically are integrated to boost performance of functions either in data centers or in embedded edge-AI devices. These edge devices commonly are battery powered and hence need to operate under constrained power budgets, mostly under a five Watt roofline. Even though the scale of work differs in edge-AI and data center paradigms, both follow similar pathway to provide efficient solutions. To improve computational throughput, most accelerators use optimization strategies which involve reduced precision arithmetic, or architectural-level enhancements, such as minimizing of data movement (through using in- or near-memory computing) and increased parallelism.

In this context, researchers proposed architectural extensions for DNN accelerators, Eyeriss v2 [61], by adding a hierarchical mesh network-on-chip to limit the costly all-to-all communication within local clusters. When processing DNN sparse input, even in compressed form, this results in a significant increase in throughput and energy efficiency. Alternatively, to handle data sparsity in DNNs, ENVISION proposes input guard memories and guard control units and a dynamic-precision SIMD architecture providing energy-precision scalability [62].

Research efforts to provide energy and intermittence-aware DNN inference and training, developed the Neuro.ZERO architecture, which is based on adaptive high-precision fixed-point arithmetic to allow for accelerated run-time embedded hardware performance [63]. Additional optimizations include tensor decomposition, pruning, and mixed-precision data representation. These improvements are mainly designed in hardware, based on neuro-inspired architectures, on CMOS, or with emerging memories.

Performance and energy-wise optimizations in DNN training have driven significant research towards investigating different numerical formats. This trend is due to the fact that microarchitectural operations on fixed-point and low-precision floating-point logic (see Figure 5) are significantly more efficient in terms of area and energy than full-precision logic (e.g., 8-bit fixed-point addition is 30x more energy efficient and 116x more area efficient than FP32 addition) [64], [65]. More recently, researchers have proposed mixed-precision format for training by using hybrid Block Floating-Point (HBFP) format, which uses 8-bit BFP for tensors in the training operations (e.g., dot products, convolutions), and FP32 for the remaining operations (e.g., activations, regularizations) [66].

FIGURE 5.

Comparative energy and area cost for different precision for 45nm technology (adapted from [64], [65]).

Show All

The support of reduced precision has fueled the recent trend in integrating DNNs also to platforms that are resource and energy-constrained such as IoT devices. Different works have shown various methods that scaled down arithmetic precision to 16-bits and even to 1-bit to optimize computation performance with minimal energy consumption [62], [67]. Contrary to early models (e.g., AlexNet, VGG) which use large number of parameters and parameters proportion of the full connection layers, modern techniques have since become popular for building compact DNNs. The key idea of these techniques is mostly based on filter decomposition for images, as shown in Figure 6, and decomposition and CNNs for time-series data [68]. These DNNs, such as SqueezeNet [69] and MobileNet [70], make a perfect fit for mobile devices and for anomaly detection in IoT monitoring [68]. DNNs today have diversified in terms of shapes and sizes that vary extensively.

FIGURE 6.

Different filter decomposition solutions.

Show All

Machine learning accelerators with small footprint have shown their benefits by achieving the combination of efficiency (due to the small number of target algorithms) and broad application scope [71]. In particular, an optimized neural functional unit (mostly in terms of memory management) can achieve a speedup of 117.87× and an energy reduction of 21.08× over a 128-bit 2GHz SIMD core with a normal cache hierarchy [71]. To optimize energy efficiency for mobile devices use, Deep Neural Processing Units (DNPU) have been proposed based on optimizing heterogeneous multi-core architecture for both CNNs and RNNs [72]. Depending on the attributes of each network, the memory architecture, data paths, and processing elements are optimized for each core. Additionally, custom separation of workload can give reduced off-chip memory bandwidth needed in a CNN. Regarding an RNN, extra multiplications are reduced via quantization table-based techniques. Developers also adopt a holistic design approach to provide low-power accelerators for accurate DNN prediction for power-constrained IoT and mobile devices, using a highly automated co-design methodology that incorporates insights and methodologies across the algorithm, architecture, and circuit levels [73]. Moreover, researchers have advocated balancing the architecture in terms of cost and returns for in-DRAM calculations to speed up DNN in mobile contexts [74]. By optimizing the systolic array on a DRAM die returns include 1.7 times TOPS, 3.7 times TOPS/W, and 8.6 times TOPS/mm2 improvement over a state-of-the–art mobile GPU accelerator, while the power consumption

reaches at most 4.4 W. With the objective of energy-efficiency and accuracy, researchers have developed both software and equivalent hardware implementation for feature extraction engine and the Decision Tree classifier [75]. As a result, they've proved that a hardware version of the hash-based feature extraction engine uses just 5.7 percent of the energy that the software version does. Lightweight classification algorithms have been tested in IoT contexts with time-series data, promising accuracy and scalability, and outperforming the commonly used 1-nearest neighbor with dynamic temporal warping [76]. Through use of fewer parameters results in lower calculation costs, which is ideal for real-time, hardware-assisted malware detection [32], [77].

In summary, a wide spectrum of prevailing techniques have rapidly enabled a new landscape for edge computing integrating ML and DNN-assisted processing.

B. ML-Based Methods for System-on-Chip Protection

Broadly, to detect patterns of abnormal behavior, various methods use memory image probing and analysis at the OS level, due to flexibility and easy access. OS-level techniques can be subject to software attacks (e.g., kernel rootkits may compromise the OS-level logging system), or even hypervisor-level forensics solutions can be the attack target itself [78]. Hence, modern methods, which establish a machine learning (ML)-based offline or runtime analysis, have shifted the initial OS-level approach so that to rely exclusively on data collected directly through the hardware. The goal is to refrain from using a hypervisor or an OS, due to credibility of the provided information, tampered by an adversary.

1) ML-Based Methods for System-on-Chip Protection From Hardware Trojans

Hardware producers frequently outsource multiple elements of their design and/or fabrication processes to keep up with the rising interest for IoT devices and the globalization of hardware fabrication. These methodologies allow harmful circuits, such as Technology Trojans (HTs), to be inserted into current Systems-on-Chip(SoCs) hardware, which is becoming an increasingly serious concern [79], [80]. HTs may leak encrypted information, degrade device performance or lead to total destruction. HTs are usually divided into four categories: (i) denial-of-service, (ii) function change, (iii) performance-degradation, and (iv) information leakage. For instance, DNN inference behavior can be successfully tampered at run-time with deliberately degradation of the victim inference accuracy through memory-efficient rowhammering and precise flipping of targeted bits [81].

Researchers primarily employ two fundamental detection and defense techniques, (i) tackling side-channel attacks via leaking of power/thermal/delay/optical/electromagnetic information, and (ii) logic testing-based, by using key-to-signature mechanisms and assuming the existence of a "golden model".

Early works used side-channel information for Trojan identification such as analysis of the path delay to generate a unique fingerprint that can be used to distinguish tampered chips [82]. Based on ARMv7 microprocessor's operating frequency deviations, by integrating analog Trojan circuit it is shown to detect an extremely rare HT, triggered by successive toggling events [83]. Additionally, the measurement of process control monitors (PCMs) was combined with a machine learning technique, a one-class Support Vector Machine (SVM), to obtain a more precise categorization boundary in identifying abusive behavior of circuits, which considerably increased the efficiency and accuracy [84]. By reverse engineering (RE), it is shown that recovered images can represent the physical structures and layout of the ICs, which are

classified based on support vector machines, particularly one-class v -SVM, to distinguish between random differences and the systematic differences caused by Trojan insertion [85], as shown in Figure 7. This means that the aim is to identify Trojans while allowing for manufacturing and reverse engineering process variances. These approaches can be paired with the usage of Deep Convolutional Neural Networks with intrinsic extraction of invariant and non-linear features to overcome manual or domain-specific feature extraction [86].

FIGURE 7.

Block diagram of one-class v -SVM trojan detection approach.

Show All

Moreover, hardware Trojan activation is proved to be successfully detected by comparison of power use between Trojan clear and Trojan embedded benchmarks via using machine learning techniques [87]. Also, by using random forest classifier, recent works demonstrate how to extract effective Trojan features from hardware-Trojan infected nets in ICs [88]. Following a hybrid approach, researchers propose to combine the signature extraction mechanism with machine learning algorithms to develop a self–learning framework, as depicted in Figure 8, that can detect the intruded integrated circuits [89]. As this researh work shows, the decision tree (DT) algorithm is the best among selected prediction algorithms (i.e., decision tree from eager learning algorithms, bayesian classifiers from probabilistic learning and k-nearest neighbors from lazy learning) in term of accuracy and precision.

FIGURE 8.

ML-based trojan detection methodology by using process and mismatch variations as timing signatures (adapted from [89]).

Show All

Additionally, to tackle complex and expensive on-chip learning-based approaches, a deep invasive methodology with a lightweight, low-power ML-based monitor for HT detection can give competitive benefits, given of a proper training dataset is utilized [90]. Alternatively, by using on-chip sensors and classification on the basis of statistical distribution of grid-partitioned power consumption, runtime Trojan detection approach gives promising results [91]. In particular, in the design phase, the ML training process uses power profiles by measuring the combined power consumption of each component involved in a particular pipeline stage along with the Trust-Hub benchmarks [92], which are then used for HT detection at runtime.

With an actually realized hardware architecture for Support Vector Machine kernel, a proposed security framework gives a detection accuracy of up to 97% for three expected Trojan attacks for a NoC-based many-core architecture [93]. The detection efficiency, in terms of accuracy (without ignoring the complexity and integration convenience), depends both on the type of the

Trojan attack and the type of the machine learning model used. Given a supervised learning model, such as SVM, DT or LR, traffic diversion attacks can be detected with an accuracy that exceeds 95%. For example, by using decision trees, core address spoofing, route looping and traffic diversion can be detected with an accuracy reaching 94%, 95% and 99%, respectively [93]. In contrast, in this category of attacks, the unsupervised learning models prove to be more deficient in terms of prediction accuracy. Figure 9 summarizes key points regarding classification of discussed strategies.

FIGURE 9.

Taxonomy of HT attack detection by ML-based methods.

Show All

It must be noted though, that there is still lack of machine learning-based algorithms for identifying the HTs compared to detecting HTs. However, the usage of classification methodology which involves machine learning for HT detection is complex and depend on the detection techniques used (e.g., shallow ML algorithms for detection are mostly target-specific and prone to underfitting or overfitting).

2) Detection and Protection Against Attacks to ML Computing

An additional direction of research involves detection, at run-time, of the correctness of a neural network's computations, such as Safe-TPU [94]. This is essentially a verifiable Trojan resilient hardware accelerator for DNNs that detects arbitrary Trojan misbehaviour, regardless of how the Trojan is designed or triggered (time-based or cheat-code based Trojans). Essentially, besides the software attacks, hardware trojans might be carefully designed to compromise the neural network's integrity, in terms of the trigger, or of the payload (i.e., the input, computational block, intermediate data and output) [38]. Modern object detection platforms, such as YOLO [95] and Mini-YOLOv3 [96] for embedded devices, expose a hardware attack surface, as shown in Figure 10, with a number of options, including:

- Model Corruption: compromise the model parameters stored in memory so that the model results deviate in all tasks
- Backdoor Insertion: alter the model itself which is stored in memory so that it provides near random results partially or fully
- Model Extraction: extract the model from the device during run-time or via proving non-volatile memory
- Spoofing: interfere and manipulate the model input data through tampering with the input sensors or with the environment
- Information Extraction: infer model information by capturing and analyzing the physical side-channels

FIGURE 10.

Attack surface of a YOLO object detection framework, composed of 24 convolutional layers, followed by two fully connected layers.

Show All

To enable accurate NN Trojan detection on resource-constrained embedded devices, recent research efforts target algorithm/hardware co-design for an end-to-end method through using a pair of input (based on Discrete Cosine Transform, DCT, extraction) and latent feature analyzers [97]. To provide strong integrity and privacy guarantees for a NN execution, authors used secure enclaves, i.e., a Trusted Execution Environment (TEE) and at the same time outsource non-critical functions from a TEE to a faster co-processor [98]. Neuron obfuscation can effectively combat increasing risks to IoT edge devices and enable security of critical data and DL model parameters, while relying on a secure key storage facility supplied by a hardware root-of-trust such as Trusted Platform Module (TPM) [99]. In IoT environment, it is important to determine adversarial attacks in real-time, attempting to compromise Network Intrusion Detection Systems (NIDS) that employ DNNs and CNNs for identifying benign from malicious network traffic [100]. In this scope, designing accelerating circuitry is an emerging topic in the deep neural networks area for security, by for example, using memristor crossbar arrays to significantly improve the throughput of the visual adversarial perturbation system [101].

C. ML-Based Methods for Embedded SoC Protection From Malicious Software

Most techniques that involve ML in device level are mainly custom specific, in the scope of the type of attack surface and of the device attributes. For instance, an ML-based approach is proposed in wireless networks-on-chip (NoCs), to identify jamming-based DoS attacks and evesdropping originating from either an internal or an external attacker [102]. They use burst error correction codes to estimate the number of burst errors in packets captured at the receive transciever. With the aid of ML classifiers (artificial neural network, support vector machine, k-nearest neighbors, and decision tree), DoS attacks are then distinguished from random transient burst errors (due to power fluctuations, ground bounce or crosstalk) and a defense unit is notified.

To protect embedded devices from malicious software components that can perform hijacking attacks in the control flow, such as code-reuse[2] attacks (e.g., like buffer overflows, return- or jump-oriented attacks) [103], designers' trend involves control flow integrity (CFI) checking. CFI examines the code execution flow graph in traces of various granularities and attests to the validity of these valid execution traces in general. However, in real-time embedded systems, especially those with restricted resources, hardware-based techniques are being developed for efficiency and resilience to software assaults, allowing for a novel way to resisting malicious software. By using the ARM CoreSight module in an ARM-based IoT environment, recent work proposes a hardware-based workload forensics framework for IoT systems [104]. By recording the spatial and temporal architecture of the address space they create a workload identification scheme that combines numerous machine learning algorithms (such as the Long Short-Term Memory (LSTM)-Recurrent Neural Network (RNN)) to assess and comprehend the workload being executed at the granularity of a process in real time. To realize anomaly detection pre-learned thresholds form the basis of comparison with the classifier outcome and thus potentially illegal program behavior is filtered out.

In general, common techniques for detecting malware or side-channel assaults rely on the use of hardware performance counters (HPC) and machine learning algorithms to build a model of the program's behavior. HPCs in multi-threaded processors are monitored in real-time to detect abnormal activity. A classifier is used to detect out-of-profile behavior by comparing retrieved characteristics to features from a previously set baseline [105]. To characterize application behavior, alternative techniques collect low-level architectural information such as profiling data from memory address references, instruction opcodes, and Translation Lookaside Buffer (TLB). However, because they rely significantly on the determinism, authenticity, accuracy, and availability of the information leveraged by hardware and software performance counters, even ML-based generated models may increase the already broad attack vector surface. Performance counters may unintentionally degrade the performance of machine learning classifiers because of data polution. All techniques presume the application's training phase is reliable, which is a prerequisite of most behavior-based intrusion detection systems. The runtime monitoring entity is therefore expected to be trustworthy and untampered with.

Another approach uses the inspection and analysis of electro-magnetic (EM) side channels to classify the kind of operations performed on a processor and so identify software execution sequences with no need to instrument the program; this information may be utilized for anomaly identification [106]. Aternative methods aim to decompose the time series to small and interpretable components, or to characterize the EM leakage of electronic devices via Fast Fourier Transform (FFT) and identify the frequencies that represent critical part of the executing program [107]. However, these techniques have drawbacks largely due to sensor noise and measurements sensitivity. Such methods for analyzing and evaluating a device's side-channel security via leakage detection, as well as standards (such as ISO/IEC 17825:2016) that provide a systematic set of leakage detection tests, have been observed to produce false positives [108]. Furthermore, finding accurate and suitable features and selecting effective parameters among many features is a difficult topic for ML to use for a high detection rate. Prior to implementing any security mechanism, one essential necessity is to eliminate any link that may transport trustworthy information from a secure region to the outside world, as this poses a risk [109].

SECTION IV.

ML-Based Security in Edge IoT Devices

Modern embedded systems inside IoT infrastructures necessitate a higher degree of dependability, accessibility, and robustness, for industrial, automotive and healthcare applications. Because traditional machine-learning approaches that run in the cloud cause reaction time delays, current innovations suggest that ML techniques and smaller-scale models will increasingly shift to edge devices, in the proximity of data sources. Big data transfers to cloud-hosted machine learning processing may cause networking flooding and large round-trip latency as compared to edge processing. Meanwhile, millions of low-cost tiny computational devices in the real world represent a significant amount of underutilized processing power. Some learning algorithms, such as instance-based learning, may, however, be too costly for edge devices. As a result, the accuracy of outcomes in IoT end-nodes may not be as great as in cloud-based systems in some circumstances.

A. ML-Based Intrusion Detection and Protection in IoT Decentralized Environments

To tackle security and privacy issues, several approaches use ML-based techniques integrated in schemes spanning end user-fog-cloud environments. Whilst conventional cloud computing

solutions might be adapted to handle some security and privacy concerns with fog computing, the latter's unique features, such as decentralized infrastructure, mobility support, location awareness, and low latency, provide unique security and privacy challenges. Because of the decentralized architecture of fog computing, it is difficult to collect and manage evidence and behavior information about fog nodes to evaluate their trustworthiness and build a trust evaluation model for all fog nodes in the network, behavior-based ML methods for increasing security and privacy in fog environments are difficult to achieve.

Fog nodes that are semi-trusted are responsible to realize a trustworthy framework to aggregate multiple sensors that is based on machine learning [57]. To alleviate cloud-based overheads, the proposed technique uses a trained model to forecast the contribution of sensor readings to the aggregate sum. Additionally, to protect the training dataset against differential assaults, this technology uses differential privacy (e.g., via introducing noise).

In a different perspective, contrary to intrusion-detection schemes defending a single domain in traditional networks (e.g., enterprise, cloud, business domain), recent strategies employ learning from various domains to identify various attacks [110]. The edge data collector is responsible for collecting the IoT data, while the edge analyzer is responsible for analyzing collected data and IoT device behavior and, the edge controller, which is based on software defined networking (SDN), is responsible for gateway configuration.

Therefore, to both optimize response time and resilience of fog layer (see Figure 11), researchers propose various orchestration techniques [111], [112], or employ machine learning-based methods in a secure-conscious manner, such as the MAPE-K model [113]. This model contains four main components: management, analysis, planning and execution. Aggregated data are partitioned and packetized depending on the data type generated from sensors, and communicated via using 128-bit AES-CCM encryption. On the basis of the type of these produced and collected data, training is performed at the cloud server and the outcome model is then executed at the edge device. However, the ML algorithm that is used to generate the model must respect the edge-device constraints.

FIGURE 11.

Improving response time in hierarchical fog-assisted computing architecture through mapping and moving functions and data to the fog layer.

Show All

Anomaly detection schemes tailored for IoT cybersecurity have also been presented through using IoT gateways to host an artificial neural network [114]. Such techniques can effectively determine correct and incorrect delay and sensor values via three-input neurons. As identified, the main challenges for anomaly detection in IoT data are quantity and heterogeneity. In the same scope, a deep recurrent neural network-based malware detection methodology for the ARM-based IoT applications has provided promising results [115] via analyzing IoT devices application opcodes. By implementing three different long short-term memory configurations, this research approach showed 98.18% accuracy to detect malware with respect to the tested data set. Additionally, to improve the trustworthiness of services in a decentralized IoT

environment, researchers proposed a reinforcement learning (RL), RL-based approach to determine the service resource allocation scheme in different time periods [116]. In all aspects of cybersecurity, by adopting a data-driven approach, anomaly detection algorithms prove to provide a valuable effective toolset. Most machine learning-based IoT approaches for malware hunting focus on energy consumption patterns [117] and application's opcodes [118].

B. Hardware-Assisted ML in IoT Devices

Devices that incorparate ML for detecting and subverting attacks commonly adopt software-based solutions, such as anti-virus applications. These solutions, though, are susceptible to high risk; sophisticated malware may be equipped with smart deviation capabilities such as obfuscation, which may be successful since traditional protection schemes mostly rely on matching patterns and signatures. The tamper-immune hardware metrics prove to be an improved security feature compared to the high-level software metrics, since software features can be jeopardized via obfuscation. Hardware-assisted ML semantically involves different methods and architectures categorized as follows.

- Hardware assistance for making ML detection more accurate, i.e., minimize false positives
- Hardware accelerators to build faster ML models and inference engines
- Machine-learning-aware and deep-learning-aware optimizations of processors (i.e., vector width improvements, SIMD instructions parallelism, low-precision FP computations) to boost the performance of a range of deep-learning applications

To optimize ML-based malware detection accuracy, recent research works propose real-time collection and analysis of hardware traces [119]. These hardware-supported instrumentation traces include (i) embedded trace buffers to collect functional values of a number of trace signals over a time window in clock cycles granularity, (ii) hardware performance counters to determine statistical behavior in terms of specific architectural features such as bus or memory accesses, cache misses, branch prediction, and (iii) Network-on-Chip (NoC) traffic to provide insight in communication patterns. Experimental results show that machine learning can be effective in malware detection by utilizing such hardware traces. In a different perspective, through using architecture-agnostic methods for forensics analysis, researchers propose to reconstruct executed workload at the granularity of a single process by using the extracted features, through minimal information obtained from the processor's translation lookaside buffer [120]. Alternatively, by exploiting hardware performance counters to collect fine-grained data for each system call of unknown programs, these unknown programs can be categorized into benign or malicious [121], [122]. The programs behavior can vary, with a significant trace comprising of thousands of system calls, while some have a short trace limited to less than a hundred system calls. To tackle such variations, captured performance counter data are reduced to a uniform dimension and then classified via decision trees, random forest, neural networks, adaboost, k-nearest neighbors with promising results in a range of fidelity [121], [123]. Even more aggresively, others introduce the use of dedicated on-chip learning controllers to perform the analysis directly in hardware, possibly even in real-time, for instance by embedding neural network or logistic regression prediction co-processor to decide based on instructions and memory access extracted features [124]. Such approaches require specialized hardware designs, but offer a low power consumption footprint with zero software interference. Since most recent processors for IoT devices are equipped with hardware

performance counters that can be used for malware detection, inexpensive methods can be employed (via using low-level hardware events) to detect threatful alterations in the firmware of embedded control systems [34]. By exploiting an augmented number of hardware performance counters with reduced accuracy, limited added value is shown for different hardware classifiers to achieve better performance, accuracy against area overhead, while the combination of classification algorithms has a good performance outcome [125]. In summary, a growing interest involves the usage of low-level microarchitectural features collected from processor's performance counter registers to implement hardware classifiers for malware discovery, with little concern of combining higher level behavior (i.e., such as operating system or network activity). This strategy offers isolation from software threats at the risk to miss new, sophisticated threats. ML-based detection models that use HPC-based approaches need to become robust against algorithm subversion attacks, especially when securing Post Quantum Cryptography (PQC) implementation on resource-constrained devices, a key requirement to maintain their integrity [126].

Today, researchers suggest relocating a classifier algorithm (such as DT) in hardware to enhance both the energy efficiency of anomaly-based intrusion detection systems for probing assaults and the restricted throughput of software in resource-constrained edge devices [127], [128]. The isolation from the software environment and intrinsic robustness of circuitry against tampering are two further advantages of mapping an ML algorithm in hardware. After contrasting several approaches (e.g., naive bayes, support vector machine, k-nearest neighbor, random forest, and artificial neural networks) for real-time performance, hardware-based classifiers demonstrate excellent performance [129], with random forest outperforming other algorithms with a maximum accuracy of 98.5 percent [130]. A full framework for deploying CNN on embedded systems has also been described, which uses a mixed pruning strategy to compress CNN models and thereby alleviate memory and performance issues [131]. However, most FPGA implementations exhibit high cost in power consumption, with some exceptions [127], which does not allow integration with microcontrollers and resource-constrainted IoT devices. Earlier, solutions presented also developing feature extraction module in hardware and the use of principal component analysis as an outlier detection method for NIDSs with detection rates exceeding 99% [132].

Towards dedicated, specialized AI-workload processors, BrainChips's Akida neuromorphic processor is a revolutionary advanced neural networking processor that brings artificial intelligence to the edge [133]. The Akida NSoC is designed for use as a stand-alone embedded accelerator or as a co-processor, while also including interfaces for ADAS sensors, audio sensors, and other IoT sensors. Moreover, NeuroEdges are devices that support the implementation of edge computing systems using neuromorphic chips, named NM500 [134], and common commercial embedded boards, however mostly targetted to face recognition [135]. Research results demonstrate considerable advantage for real-time computations, thus savings in terms of the burden of requiring many datasets for effective training.

In the scope of bringing AI at the edge, IoT devices are also emerging with hardware support. Recently, ARM introduced enhancements towards boosting ML processing on top of the ARM Cortex-M55 processor, that can be up to 15 times faster than the previous version, and ARM Ethos-U55 NPU, the first micro- Neural Processing Unit, micro-NPU, for Cortex-M architecture, which can speed up ML performance by up to 480 times [136]. By integrating Deep Learning Accelerator (DLA), NVIDIA DRIVE AGX Xavier can deliver an incredible 30 TOPS for automated

driving [137]. To enable real-time sensing with limited energy generated by energy harvesting, Renesas embedded AI (e-AI) [138] demonstrated power efficiency of 8.8 TOPS/W [139]. The Renesas accelerator developed a processing-in-memory (PIM) architecture, an increasingly popular approach for AI technology, in which multiply-and-accumulate operations are performed in the memory circuit as data is read out from that memory.

In addition to emerging devices with AI-oriented hardware extensions, modern ML tools oriented to help in running AI algorithms on microcontrollers, facilitate inferencing based on models trained with TensorFlow, Keras, PyTorch, Caffe and others [140], [141]. The application code can directly use these kernels to realize neural network models on ARM Cortex-M CPUs. Moreover, developers deliver microcontroller optimized libraries, such that neural network inference can achieve 4.6X improvement in runtime/throughput and 4.9X improvement in energy efficiency [142]. First, these optimized functions accelerate key neural network layers, such as convolution, pooling and activations. Second, the optimizations aim to reduce the memory footprint, which is key for memory-constrained microcontrollers. Alternatively, these kernels can be used as primitives by machine learning frameworks to deploy trained models.

STM32Cube also helps with the easy integration of standard AI algorithms in microcontrollers. Automatic conversion of pre-trained neural networks and integration of the resultant optimized library into the user's project are made possible by the particular AI ecosystem. Cube.AI tool offers not only mappping a neural network on an STM32 MCU but also optimizations. For instance, the code generator opts for folding some of its layers and reducing its memory footprint. In particular, to optimize for condition monitoring and anomaly detection, and hence reducing anomaly detection time, STM's FP-AI-NANOEDG1 manages sensor input data collection, on-device learning sessions and inference models in real-time [143], [144]. These tools claim to make it easier to create machine learning libraries that include both inference and edge training. The purpose is twofold. First, predictive maintenance is seamlessly enabled. Second, for assault detection, sensor patterns are used in a self-learning, simplified method. The requirement for extensive knowledge in machine learning, data science, or developing neural network models is becoming obsolete as a result of tool automation. At the same time, FP-AI-NANOEDG1 offers coverage of the entire development of the machine learning cycle. This means, it helps from the data set acquisition up to generating libraries by the NanoEdge AI and integrating the application on the physical node, as well as the security and detection with sensor patterns self-learning and self-understanding. Essentially, an STM32L4R9ZI ultra-low-power microcontroller supports all tasks, data collection, learning session and real time inference, while processing physical sensor data as input.

Advancing both signal processing and neural network applications to edge-devices are also emerging for new embedded platforms that integrate multiple cores in parallel, such as GreenWaves Technologies GAP-8 and GAP-9 (i.e., nine RISC-V cores) [145], to enable embedded machine learning in battery-operated IoT sensors, mainly focusing image processing domain. Such systems-on-chip (SoCs) are among the most advanced low-power edge nodes available in the market, embodying the PULP architectural paradigm with DSP-enhanced RISC-V cores, while frameworks have been developed exploiting SoCs features, such as hardware loops, post-modified access LD/ST, and SIMD instructions down to 8-bit vector operands [146], [147]. Additionally, to provide agility for a variety of different neural network techniques, a novel domain-specific Instruction Set Architecture (ISA) for NN accelerators, called Cambricon, has been proposed [148]. This is a load-store architecture that integrates scalar, vector, matrix, logical, data transfer, and control instructions, based on a comprehensive analysis of existing

NN techniques. In summary, Figure 12 gives an overview of different directions in bringing ML processing at the edge for efficient data processing in secure manner.

FIGURE 12.

Classification and comparison of hardware-assisted ML-methods towards IoT embedded processing and security.

Show All

In conclusion, hardware specialization is a popular approach to accelerate the computation of neural network-based applications. Besides neural network and deep learning (DL) accelerators, specific microarchitectural techniques and even software methods succesfully present high performance and attempt to save energy. Microcontrollers can also provide hardware and software support for low-precision computing [149]. Research results via lower precision fixed-point arithmetic are promising in terms of memory footprint, inference time and power efficiency (by using TensorFlow Lite for microcontrollers, STM32Cube.AI and a custom tool MicroAI) [150]. DNNs also perform computations with other patterns, such as sparse lookups, vector operations and deconvolution [151]. Future CPUs will also host dedicated DL accelerators to accelerate not only such operations but also crypto- and analysis functions, thus, bringing all worlds together to resource limited devices. Vendor-optimized libraries will remain essential to leverage all the performance capacity from a processor.

However, most of these machine learning solutions mostly focus on sensor data fusion and help ecosystem to advance the future of automotives, smart buildings and wearable computing, but rarely consider anomalous components behavior in an IoT infrastructure as a prime goal. On the other hand, embedded hardware security in IoT infrastructures is a necessity to protect the identity of devices, to secure the trusted execution of their applications against tampering, and to protect the privacy and security of data they generate. Nevertheless, protection techniques such as HSM and TPM enhancing hardware security are scarcely linked to ML and DNN methods in this scope.

In addition, while a variety of useful mechanisms to protecting against memory vulnerabilities at run time have been presented, such as fine-grained tagged memory systems [152], support of pointer authenticity [153], and hardware-assisted scope enforcement [154], they are seldom integrated with ML-based solutions.

C. Secure ML and DL Inferencing Using Trusted Execution Environments

Different methods guarantee the secrecy of the assets engaged in ML computations in untrusted computing environments to ensure resilience of ML models and derived services against malicious actors. The protected assets can comprise data, machine learning models, and computation results that can compromise the confidentiality of the protected assets indirectly. In this context, researchers suggested employing trusted enclaves to offer data integrity inference, or applying privacy-preserving algorithms with the help of ARM TrustZone to safeguard peripheral access for ML data privacy [155]. Thus, even if the adversary has complete control over the software running in the user's device's normal world, including privileged

software like the commodity OS, the enclave with the ML model is attested by a SANCTUARY core [156], which creates and securely stores a cryptographic hash of the enclave's initial memory content.

Designs for DL model calculations employing tiny TCB size and limited secure memory are demonstrated for mobile and IoT devices by using the benefits of TEEs and appropriate device accelerators [157]. The supplier is required to present the genuine DL model's cryptographic hash to authenticate the DL model's integrity. Sensor data is also safely fed into the TEE using secured drivers, while encrypted data is decrypted inside TEE and sent to a protected accelerator at each stage of the inference process (e.g., a GPU), which is splitted to minimize the code base inside the trusted enclave [157], [158]. Similarly, only the most vulnerable layers of a DNN are concealed inside the TEE to prevent inference attacks [159].

TEEs have memory constraints, and the move from the untrusted domain to the TEE adds overhead, as shown in research that assessed TEE performance characteristics proving TEE-based functionality to be expensive to both invoke and execute [160], [161]. However, various promising appoaches offer significant benefits for ensuring trusted ML model execution, such as cancelling inherent memory limitations thus allowing to securely run complex models [155], or, providing better performance for protecting ML services [98], or, removing the software layer to enable a secure OS and enabling more trusted applications to run at once [162].

SECTION V.

Discussion

This section presents key points and challenges regarding cross-cutting directions as surveyed in prior sections.

A. Challenges for Realizing ML With Hardware Support at the Edge

From the perspective of the hardware architecture for DNN and machine learning, modern realizations are emerging but are mostly application-oriented. In terms of inferencing, no one architecture appears to stand out in terms of delivering critical machine learning hardware primitives to serve a wide range of applications, particularly at the edge. The field of machine learning is still in its early stages, while promisingly inferencing is shown to perform on the microcontroller by a variety of embedded systems. Provided light ML is satisfactory, such as in keyword spotting, or use-cases where response time is not critical, such as analyzing offline photos, then the microcontroller is capable of performing at such scale. A promising solution involves the realization of ML-based malware classifiers in microprocessor hardware with significantly reduced overhead as compared to the traditional software-based methods [32].

Despite the fact that early innovations utilised GPUs, which enabled a big leap forward in AI capabilities, power consumption is an important consideration in IoT devices. Inferencing via using Tensorflow models on mobile devices consumes more than the half of the consumed energy (57%) for data movement [163]. Hence, researchers propose new architectures, mostly based on processing-in-memory organizations (PIM) and small fixed-function accelerators (PIM accelerators), such as data packing and quantization to make machine learning inferencing more energy-efficient [164].

With regard to ML for secure IoT infrastructures, most hardware-oriented works mostly use Snort rules [165] rather than ML-based anomaly detection; advancements though recently demonstrated feature extraction algorithms which are suitable for hardware implementation

and promising results of feature selection methods with two simultaneous objectives, accuracy and energy consumption [75]. Additionally, for different cyberphysical systems, different compatible neural network architectures should be adopted [17]. Further, ML could play a significant role in enabling asymmetric elastic cryptography in IoT but there are challenges that need to be addressed [166], such as IoT-based anomaly datasets, probability and exact threat identification, authentication of the training data sets, zero-day attacks and real-time firmware updating of millions of devices.

B. ML-Based Secure Processing in Hierarchical IoT Environment

If workloads become heavier (e.g., big data in industrial applications, biomedical imaging, genomic systems), and where performance is critical or power efficiency is a concern, then different solutions appear for IoT resource-constrained devices, ranging from microarchitectural support for AI (i.e., at instruction level [140] and accelerators [143]), to fog-oriented solutions for ML-based applications and anomally detection systems. Fog computing, as a rising computing paradigm along with SDN and NFV technologies, can become a powerful solution in securing a variety of connected industrial environments [167]. Despite the abundance of huge data in the vicinity of loT, creating and deploying strong attack detection systems for loT devices is difficult due to resource limits, latency sensitivity, and distribution concerns [168]. As fog computing provides a distributed environment with multiple fog nodes near to IoT devices in the edge layer, recent methodologies have demonstrated the usefulness of LSTM-based DL models in cybersecurity to identify a variety of threats with high detection and accuracy rates [169]. However, implementing a heavy DL detection solution directly on low-capacity IoT devices (to detect even morphing attacks), detecting multiple threats with high detection rates and accuracy rates, and monitoring and updating the detection system to identify new attacks remain difficult.

In the scope of improving ML-based IDS at IoT system level, in particular to address the strict latency requirements that challenge the detection of cyber-attacks, alternative proposals include a fog architecture to benefit from the low latency provided by fog nodes [31]. Further, hardware support for ML inferencing is deemed important for real-time IDS methods. However, as researchers show [170], program behaviors tend to deviate at an early stage of their execution and may therefore be benefited to perform the real-time monitoring and identification analysis using hardware techniques as well [170], [171].

C. Methods AND Tools Support for Realizing Automated and Trusted ML-Based for IoT Devices

As the EDA tools and methods keep evolving, the implementation of hardware IDS methods in IoT devices on the basis of machine-learning algorithms and even with Trojan security aware methods [172], are increasingly boosted. This is facilitated by the wider acceptance of continuously more efficient high-level synthesis tools, based on widely known and used languages, such as OpenCL, C/C++, or MATLAB, thus enabling software designers to take advantage of FPGA technology [129], [173]–[175]. On top, with the goal to provide complete end-to-end toolchain to empower domain scientists to design machine learning algorithms for low-power devices, new developments are presented for a range of devices [176].

Additionally, in adversarial environments which are inherently non-stationary, such as the cyber security domain, ML/AI-based IDS methods and security critical applications require further advancements in terms of reliability to address adversarial machine learning (e.g., machine

learning poisoning of training data-sets and attack models) with degrading sub-optimal decisions, thus resulting in endless cyber-war gaming between defense and attack strategies [26], [49], [177], [178]. Finally, an orthogonal line of research should pursue protection of both IoT device and system secrets, even in the presence of compromised system software layers and malware.

IDS and defences developed to protect from adversarial examples, have shown great accuracy through employing DNN methods, but also a wide space for parameters tuning and reduced robustness due to adversaries capacity to evade even if an adversary is oblivious to a specific defense [179]. Essentially, an important challenge today involves not only designing accurate DNN-based defense schemes, but additionally producing interpretable results in terms of understanding how the ML algorithms reach into the conclusion for detecting attacks. Complementary to efforts towards mitigating false classification by augmenting training data for compensating undertraining, new schemes propose increasing quality of explanations for individual classification outcomes for security applications [180], thus raising the trust of users.

Furthermore, the defense strategy should advance its understanding of the pathway and parameters to generate a non-binary detect decision, jointly with how an attacker might react to any defense. The protection scheme needs to ensure that the defense remains secure against an attacker who discovers how the defense works.

SECTION VI.

Conclusion

Machine learning and edge computing solutions promise to efficiently distribute the processing needs across devices, servers, and gateways so they can act on sensors data from heterogenous devices in real time and predict outcomes locally. It is also widely recognized that employing machine learning and neural network-based methodology is able to overcome quantity and heterogeneity challenges of IoT devices and data in detecting anomalies in the data sent from edge devices (through for example focusing on behavior and protocols). However, machine learning and deep learning algorithms are generally computationally and memory intensive, making them unsuitable for resource-constrained environments such as IoT, mobile, devices and gateways. To efficiently implement these compute and memory-intensive algorithms within the IoT computing space, especially in terms of energy requirements [75], innovative optimization techniques are required at the algorithm and hardware levels.

Tradeoffs between specialized processors and general-purpose processors will continue to confound the industry for the foreseeable future. This may provide an opening for new technologies, memories, eFPGAs or other programmable logic or software, but there is still a long way for a solid ground in confident industry adoption. Security countermeasures should elevate as a first class constraint, moving from a subsequent concern in IC design, contrary to traditional goals involving cost, performance, and reliability [181]. Additionally, the tradeoff for opting for the best solution through a purpose-built processor for efficiency, or through an off-the-shelf component will vary widely by application and ultimately by how these solutions perform over time and under load. Regardless, the inferencing market has opened the door to much different architectures and approaches than in the past, and there is no indication that will change anytime soon.

The rise of Internet of Things and edge systems and their use in large-scale, commercially sensitive applications makes attacks a growing concern for developers in all application

domains. Many mitigation techniques come with major overheads in power performance and silicon die area that are impractical for IoT devices. On top, a growing concern involves how machine learning assists in securing IoT infrastructures, or if deep learning reverses the effects of countermeasures. Nevertheless, as machine learning based IDS obtained using hardware acceleration, compared to software, reaches high levels of accuracy, more that 95%, and boosts the classification speed significantly, open the way for integration at the IoT edge devices, which is especially challenging in real-time applications. Additionally, anomaly detection in IoT systems with transient behavior and, in domains that rapidly evolve becoming smarter (e.g., vehicles becoming more intelligent), is highly important and challenging, ultimately needing designs of effective and proactive secure IoT infrastructures. Moreover, it is also important to focus on developing IoT protection mechanisms that detect known and unknown attacks while being protocol-independent and non-cryptography related. Concentrating on the challenge of exploding unlabelled data in IoT and developing labelled IoT datasets for anomaly detection purposes, are also important research areas.

Hopefully, this article will be useful for academia and industry research, to identify the advantages and security drawbacks of different machine learning methods for an IoT infrastructure. Additionally, this survey will enable security and privacy designers enhance IoT devices countermeasures from traditional ones, while unleashing the development of efficient, low-latency, and reliable, ML-based intelligent services.