future internet

MDPI

Review

A Survey on Edge Computing (EC) Security Challenges: Classification, Threats, and Mitigation Strategies

Abdul Manan Sheikh [1,2,*], Md. Rafiqul Islam [2], Mohamed Hadi Habaebi [2,*], Suriza Ahmad Zabidi [2], Athaur Rahman Bin Najeeb [2] and Adnan Kabbani [1]

[1] Department of Electrical Engineering and Computer Science, College of Engineering, A'Sharqiyah University, Ibra 400, Oman; kabbani_a@asu.edu.om

[2] Department of Electrical Computer Engineering, Kulliyyah of Engineering, International Islamic University, Kuala Lumpur 53100, Malaysia; rafiq@iium.edu.my (M.R.I.); suriza@iium.edu.my (S.A.Z.); athaur@iium.edu.my (A.R.B.N.)

* Correspondence: abdul.manan@asu.edu.om (A.M.S.); habaebi@iium.edu.my (M.H.H.)

Abstract: Edge computing (EC) is a distributed computing approach to processing data at the network edge, either by the device or a local server, instead of centralized data centers or the cloud. EC proximity to the data source can provide faster insights, response time, and bandwidth utilization. However, the distributed architecture of EC makes it vulnerable to data security breaches and diverse attack vectors. The edge paradigm has limited availability of resources like memory and battery power. Also, the heterogeneous nature of the hardware, diverse communication protocols, and difficulty in timely updating security patches exist. A significant number of researchers have presented countermeasures for the detection and mitigation of data security threats in an EC paradigm. However, an approach that differs from traditional data security and privacy-preserving mechanisms already used in cloud computing is required. Artificial Intelligence (AI) greatly improves EC security through advanced threat detection, automated responses, and optimized resource management. When combined with Physical Unclonable Functions (PUFs), AI further strengthens data security by leveraging PUFs' unique and unclonable attributes alongside AI's adaptive and efficient management features. This paper investigates various edge security strategies and cutting-edge solutions. It presents a comparison between existing strategies, highlighting their benefits and limitations. Additionally, the paper offers a detailed discussion of EC security threats, including their characteristics and the classification of different attack types. The paper also provides an overview of the security and privacy needs of the EC, detailing the technological methods employed to address threats. Its goal is to assist future researchers in pinpointing potential research opportunities.

Keywords: edge computing; cloud computing; data centers; bandwidth; artificial intelligence; PUFs

Academic Editors: Jordi Mateo-Fornés and Choong Seon Hong

Received: 15 February 2025
Revised: 1 April 2025
Accepted: 12 April 2025
Published: 16 April 2025

Citation: Sheikh, A.M.; Islam, M.R.; Habaebi, M.H.; Zabidi, S.A.; Bin Najeeb, A.R.; Kabbani, A. A Survey on Edge Computing (EC) Security Challenges: Classification, Threats, and Mitigation Strategies. *Future Internet* **2025**, *17*, 175. https://doi.org/10.3390/fi17040175

1. Introduction

The Internet of Things (IoT) is a comprehensive network of interconnected physical devices equipped with sensors, software, and various communication technologies, enabling them to communicate and share data over the Internet. Powered by smart devices, edge computing (EC), and big data analytics, IoT is transforming both business operations and the interactions between service providers and customers [1]. The number of IoT devices is estimated to nearly double, increasing from 15.9 billion in 2023 to over 32.1 billion by 2030. IoT-based services are rapidly being adopted across various industries and consumer

markets, including healthcare, industrial automation, automotive, smart cities, logistics, and agriculture [2,3]. The adoption of IoT has been further driven by the integration of advanced technologies such as 5G, AI, Blockchain, and EC [4,5]. A typical IoT architecture, shown in Figure 1, consists of devices, sensors, and actuators in the perception layer, generating a large amount of data which requires further processing to enable intelligence for service providers and end users. The network layer transports data from the perception layer to the network via gateways that might perform preprocessing and often gather data from other edge devices. Authentication, encryption, malware protection, processing, and initial decision-making are carried out in this layer. The data processing is either carried out in the gateway or the cloud, based on the application and implementation. Cloud computing technology requires sensed data to be uploaded to centralized servers called data centers for further processing, and the results are transmitted back to the device layer. Such a centralized processing approach puts enormous pressure on the communication network regarding bandwidth, latency, and the vulnerability of data security [6].

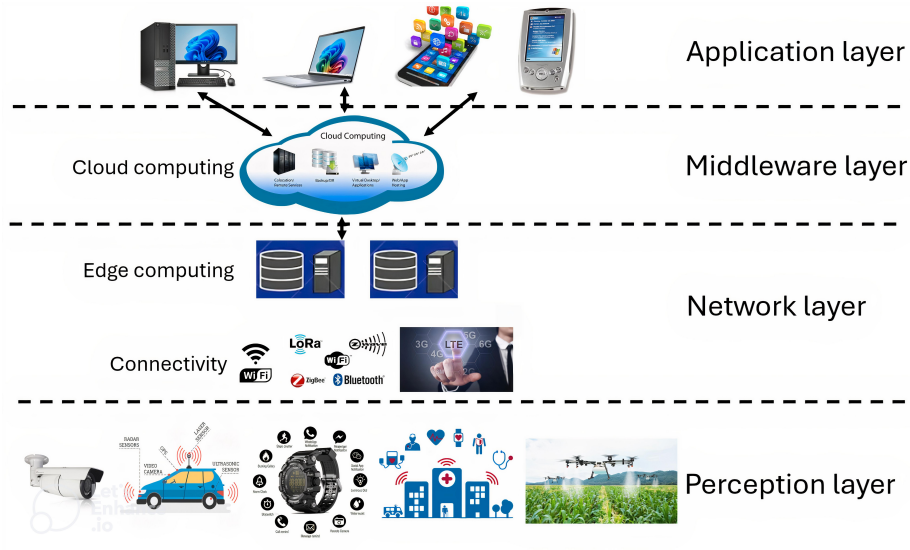

Figure 1. A typical IoT architecture.

Centrally located servers at data centers offer poor quality of service (QoS) in addition to the burden imposed on the communication networks, including the following:

- Additional costs are involved due to inefficient utilization of bandwidth and network resources;
- Large-sized data drastically degrade network performance;
- Billions of connected devices on the IoT network make it difficult to manage data traffic; and
- Time-sensitive IoT applications are bound to become affected due to network-introduced latency [7,8].

The idea behind "edge computing (EC)" is to minimize communication latency and bandwidth usage, enable real-time data analysis, reduce operational costs, enhance scalability, and improve service quality [9–12]. Closer EC proximity to the data sources reduces transmission delays, packet loss, and high energy consumption [10,13,14]. Additionally, EC offers location-aware services and enhances resource allocation by shifting tasks from IoT devices with limited resources to more powerful edge servers [15]. Thus, EC is characterized by its heterogeneous distributed network architecture, large-scale data processing, parallel computing capabilities, and support for mobility services, including location tracking. However, EC increases vulnerability to cyberattacks and threats, as sensitive data are

stored and processed in a distributed environment with limited resources, making it difficult to implement complex security algorithms [7,16]. Figure 2 shows an edge-based IoT attack model outlining various threats and vulnerabilities specific to the edge computing environment in the Internet of Things (IoT) context [17].

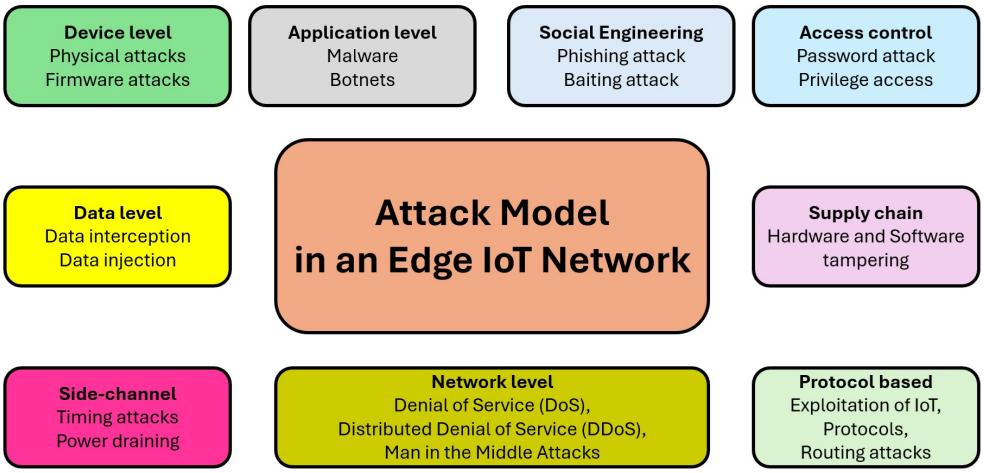

Figure 2. IoT attack model.

Additionally, the dynamic nature of the edge in IoT networks makes them more susceptible to security attacks and difficult to protect. Mostly, the data security threats and attacks on EC architecture are placed under four categories, i.e., distributed denial-of-service (DDoS) attacks, side-channel attacks, malware injection attacks, and authentication and authorization attacks [18]. Xiao et al. provide a classification of security attacks in an edge environment under six categories, i.e., DDoS attacks, side-channel attacks, malware injection attacks, authentication and authorization attacks, man-in-the-middle attacks, and bad-data injection attacks [18]. A threat intelligence report from "Netscout" reports an upsurge in DDoS attacks during the second half of 2021. About 9.7 million attacks were identified in 2021, which is 14% higher than in 2019 [19]. The number of malware attacks on IoT devices has grown from 813 million to 2.9 billion from 2018 to 2020 [20]. Current research on EC security and privacy is focused on techniques such as data privacy, lightweight security protocols, artificial intelligence (AI) integration, trust management, and collaborative security. Differential privacy (DP) adds noise to data to protect individual privacy while allowing aggregate data analysis in five critical areas: data transmission, data processing, data model training, data publishing, and location privacy [21–23]. Authors of [24] introduce a hybrid differential privacy model combined with adaptive gradient compression, providing stronger protection against inference attacks while transmitting gradient parameters. Implementing secured lightweight encryption and authentication techniques secures data from side-channel and hardware attacks [25]. Samad et al. proposed an anonymous authentication protocol that utilizes elliptic curve cryptography (ECC) and signcryption techniques [26]. Several encryption models have been developed over the years using or combining various techniques like authenticated encryption (AE) with associated data (AEAD) schemes [27].

Trust management is a critical component of EC, involving the processing and storage of data at the network's edge. Blockchain and distributed ledger technologies (DLTs) enable decentralized, secure, and transparent trust management. Wang et al. introduced a blockchain-based secure data aggregation strategy (BSDA) integrating a security label into the block header, which includes the task's security level (SL) and completion requirement (CR) [28]. A blockchain-based protocol introduced in [29] supports conditional anonymity and efficient key management, overcoming the limitations of traditional cryptographic pro-

tocols. Authors of [30] propose multiple edge blockchains that interact through a cloudlet chain operating independently. Collaborative EC enhances privacy by selectively sharing data or insights among nodes or with the cloud, minimizing overall exposure to sensitive information. Techniques such as federated learning (FL) allow multiple devices to train ML models locally on their data without transmitting the raw data to others. This approach helps organizations comply with data residency and privacy regulations by ensuring that data remain within designated geographic boundaries. Li et al. have developed algorithms based on Multi-Armed Bandit (MAB) frameworks by sharing information about server security risks [31] while proposing an SDN-based framework [32]. Data disturbance and adversarial training methods are adopted in [33] for generating adversarial samples using the Firefly Algorithm (FA).

AI-driven techniques enhance intrusion detection, data confidentiality, and access control in edge environments. Researchers have developed various strategies to secure EC, including machine learning (ML) algorithms and innovative methods like hybrid feature analysis. ML is especially effective in detecting real-time anomalies and potential breaches, offering robust protection against advanced attacks [34,35]. AI chips with computational accelerators like Field Programmable Gate Arrays (FPGAs), Graphics Processing Units (GPUs), Tensor Processing Units (TPUs), and Neural Processing Units (NPUs) are integrated into intelligent mobile devices [36]. Field programmable gate arrays (FPGAs) are suitable for implementing customized hardware logic and real-time image processing for high-performance edge computation [37]. FPGAs' characteristics suit EC requirements like (i) processing of data streams at lower latency, (ii) adaptability to any algorithm due to their reconfigurable architecture exploiting spatial and temporal parallelism at a finer granularity, and (iii) thermal stability reducing cooling cost [38]. FPGA-based edge devices have proven their resilience to physical and side-channel attacks. FPGAs' inherent ability to process tasks in parallel and flexibility in handling diverse workloads can match AI and ML algorithms' computational and processing needs. FPGAs allow greater flexibility in what the processor does, they are very useful in building AI accelerators [39]. The FPGA-based edge reduces the response time by 1.62× for the object application and 1.14× for the face application compared to CPU-based edge offloading in general [40]. Zhao et al. presented a novel approach to secure FPGA-based edge devices using a lightweight hardware-assisted chaos-based stream cipher for protecting FPGA bitstreams [41]. Regarding security, IP protection techniques implemented on FPGA have better flexibility and require no extra resource overhead compared to those implemented on a traditional custom circuit. Ngo et al. implemented a hierarchical decision-making approach combined with an ANN model as a hardware-accelerated framework on the FPGA for real-time detection of network intrusions [42]. An Oscillator Collapse (OC-PUF) designed to utilize manufacturing variations in FPGAs that generate unique responses to input challenges was tested on Altera DE2-115 FPGA boards, achieving an inter-chip Hamming distance of 46.7% [43]. FPGAs can run several lightweight cryptographic protocols simultaneously, in addition to advantages like optimal chip area, speed, and power consumption [44]. Silicon physically unclonable functions (PUFs) implemented on FPGA platforms are flexible, secure, cost-effective, and offer a quick turnaround. FPGA-based PUFs are diverse and effective in IP protection [45], RFIDs [46], secured key generation [47], and remote activation [48].

The remainder of this paper is divided into the following sections. Section 2 discusses the basic edge IoT architecture and key components of the edge ecosystem across three distinct layers of cloud, edge, and devices. Section 3 comprehensively discusses security, privacy challenges, associated countermeasures, and defense mechanisms deployed in an edge paradigm. Section 4 provides the implementation details of PUFs for device-specific authentication schemes in hardware security, digging into the reasons responsible for

authentication and trust challenges, access control, and root causes of edge computing security threats, and also proposes future research directions. Finally, we conclude this paper in Section 5.

1.1. Motivation

The primary strength of EC in IoT network security lies in its decentralized architecture. EC reduces the need for data to travel, thus reducing the potential attack surface. Hence, due to its precise control over data processing locations and methods, the EC can uphold data privacy regulations like the General Data Protection Regulation (GDPR) or the California Consumer Privacy Act (CCPA). Therefore, EC enables the implementation of security protocols and analytics directly at the device level, facilitating real-time threat detection and response. EC is expected to possess flexible, lightweight, secure, and self-adaptive data analytics mechanisms based on user data sensitivity. However, the decentralized architecture of EC poses significant trust management challenges due to the issues related to collecting and managing evidence information from edge devices [49]. ML-enabled EC can make data-driven inferences, predictions, and decisions based on acquired knowledge from past data. Also, ML is a preferred choice for IoT services' privacy and data security due to its analytical capability. However, ML-based security schemes possess serious limitations due to the need for huge training datasets and privacy issues. So, there is an urgent need to devise an ML-based security scheme with low computation and communication costs [50]. ML-based modeling through anomaly detection techniques offers more generic and robust security solutions against unknown attacks. However, ML-based security solutions may be vulnerable to ever-evolving attacks like adversarial ML [51]. With a large amount of data generated by edge devices, there is a need for ML models that can run on resource-constrained edge devices. Also, techniques should exist for compressing the ML models that can make these models lighter and faster while making them suitable for edge deployment. Traditional security solutions rely on cryptographic methods where a secret key is stored within the device. However, the entire security system can be breached if this key is compromised. In contrast, a Physical Unclonable Function (PUF) uses the hardware itself as the medium to generate a unique secret key. The core principle of PUFs is based on the unique, device-level variations introduced during the manufacturing process. PUFs can be applied in various security protocols, including unique identifiers, secret keys, device authentication, intellectual property protection, and pseudo-random bit generators (PRNGs) [52].

1.2. Comparison with Existing Literature

Table 1 lists recently published research articles that comprehensively survey data security and privacy challenges and their mitigation techniques in the context of EC-based IoT services. Topics covered include comprehensive trust management frameworks, mechanisms orchestration, and standardization, software-defined networking (SDN), blockchain, ML techniques, and diverse versions of cryptosystems. PUFs enable the authentication of integrated circuit (IC) chips by exploiting inherent device variations. These features include random delay characteristics of wires and transistors due to process variations during semiconductor manufacturing processes [53]. PUF-enabled RFIDs and processors are under development that can generate cryptographic keys and make physical cloning of semiconductors difficult [47]. Majzoobi et al. published a survey on PUF-enabled security primitives for field programmable gate arrays (FPGAs) that can mitigate IP theft and tampering at HDL, synthesis, and bitstream levels [54]. Edge machine learning (ML) implementation models and architectures were surveyed by Merenda et al. [55]. Edge ML effectively reduces data load on the IoT network while improving privacy. Also, various

security aspects, effective countermeasures through edge artificial intelligence (AI), and the potential to improve edge AI through blockchain and Deep Reinforcement Learning are highlighted in [56,57].

Table 1. Existing surveys on EC security.

Reference	Scope	Focus	Limitations
[7,9,55,58–62]	Review of Opportunities and Challenges in EC	Conversations on EC-assisted IoT architectures, data security, and privacy-related challenges alongside insights into potential future research directions. Implementation of AI/ML-assisted cryptography algorithms and protocols is crucial for ensuring reliable access and control over the network, storage, and computation across numerous distributed edge nodes.	Limited resources at edge devices act as a barrier in terms of scalability and flexibility issues. Also, cryptography protocols have difficulty protecting endless data streams or as the data arrives [63].
[16,64–66]	Network security architecture	Secure access service edge (SASE) network architecture integrated with Virtual Private Network (VPN) and software-defined wide area network (SD-WAN) characteristics ensures secured web gateways, firewalls, and zero-trust network access.	Converging network access and security into a single network architectural model may be a challenge.
[17,47,54,67–69]	ML and deep learning (DL) models in the context of Edge security	Discussion on centralized, decentralized, and hybrid architectures implementing AI at the edge as well as technologies like federated learning, model compression, gradient compression, DNN splitting, and gossip-based training.	Maintaining and updating the ML models over time and training on the cloud.
[70–74]	Intrusion detection system	Host-based intrusion detection systems (HIDSs) monitor individual devices, while network-based intrusion detection systems (NIDSs) analyze network traffic for potential threats.	The limited computational and storage capabilities of edge nodes limit the processing or storage of large-scale data.
[75–78]	PUF-enabled digital fingerprint	PUFs utilize the distinctive physical traits of edge devices to offer robust authentication, secure key management, and tamper resistance while eliminating the need for stored cryptographic keys.	Highly sensitive to environmental factors like temperature, voltage, and electromagnetic interference, PUFs exhibit unique challenge–response pairs (CRPs) and are vulnerable to machine learning attacks.

Ref. [79] proposes a blockchain-enabled FL-based architecture that integrates blockchain technology with FL for decentralized training and secure data exchanges in UAV networks. The authors of [59] carried out cryptanalysis of a blockchain-based decentralized security solution for EC, i.e., DecChain architecture, using the AVISPA simulation tool. The authentication and transactions between users and service providers are verified through blockchain mechanisms. An overview of the integration between blockchain and EC systems, providing a tamper-proof transaction ledger, is discussed in [61,80]. The survey identifies the critical issues in areas such as scalability, self-organization, security, resource management, and combining blockchain's consensus algorithms with EC's dynamic nature. Zhao et al. conducted a detailed study on the benefits of integrating EC with cloud computing and performance issues related to resource management, virtualization, and networking in several

sub-aspects [69]. Authors of [71,74] review the integration of intrusion detection systems (IDSs) and ML techniques on known signatures for an adaptive and efficient performance. A Hybrid Intrusion Detection Framework (EHIDF) for addressing security threats in Mobile Edge Computing (MEC) is proposed in [73], utilizing modules like Signature Detection Module (SDM), Anomaly Detection Module (ADM), and Hybrid Detection Module (HDM). The framework was evaluated using the UNSW-NB15 dataset, which includes various attack types. The researchers in [75] implemented a double PUF-based model on the Xilinx Virtex5 FPGA for authenticating edge devices and software (IP cores), achieving a 61.96% reduction in resource utilization along with a performance stability rate of 99.54%. In [76], XORArbiter PUFs were used for authenticating Edge Data Centres (EDCs) and edge devices, with Raspberry Pi devices simulating EDCs. In [77], a 10-transistor SRAM cell was utilized to perform both XOR encryption (PUF) and MAC operations within the same cell, allowing the processing and encryption of DNN model weights. A delay-based PUF, producing a 1-bit signature, was synthesized and configured on a 28 nm FPGA using on-chip resources such as lookup tables (LUTs) and flip-flops, achieving an average uniqueness of 49.7% [78]. A fast and effective data encryption application, called Selective Encryption and Component-Oriented Deduplication (SEACOD) [81], is discussed in context to the EC security [82]. A blockchain-based mutual authentication scheme integrated into certificateless cryptography, elliptic curve cryptography, and pseudonym-based cryptography that authenticates transactions between edge servers and IoT devices is presented in [83]. Also, the key generation negotiation mechanism while considering IoT devices' mobility is implemented on hyperledger fabric. A review of the current research status in EC security on access control, key management, privacy protection, attack mitigation, and anomaly detection is carried out in [84]. The authors advocate the need for innovative proposals in EC, as already mature cloud computing does not meet recent challenges and requirements. Access control and key management schemes in Information-centric networking (ICN) and non-ICN infrastructures are based on traditional schemes, and there is a need for newer architectures with lighter encryption protocols. A secure searching scheme for desired data within own/shared data on storage, as well as a searching scheme for IoT smart devices at the edge of cloud-assisted IoT, is proposed in [85]. The researchers claim that their proposed data-sharing mechanism, along with secret and public key encryption, improves data processing time as compared to existing cloud-based systems.

A comprehensive overview of blockchain technology and its application in the network control, storage, and computation at edge nodes, offering network security, data integrity, and computation verification, is presented in [60,61,86]. A blockchain technology integrated into the communication layer of an edge network can manage the radio spectrum and authentication of edge devices, as well as network access control in the network layer [86]. Liu et al. proposed blockchain-based data and energy coins on the distributed consensus principle for the secured data exchange in Electric vehicles cloud and edge (EVCE) computing [58]. Blockchain-based decentralized framework named "DecChain" is proposed to eliminate the need for authentication to access third-party services or resources [59]. Also, hardware-assisted blockchain implementation of a defense-in-depth strategy and proper network segmentation forms the basis for a secured and trusted environment for the unidirectional payment channels is investigated in [62,64]. Infrastructure for cloud access through the adoption of the Secure access service edge (SASE) framework is used for developing strategies for threat and intrusion detection, network segmentation, and defense in depth (DiD) [65,66].

1.3. Novelty and Contribution

The geographical distribution of edge devices increases the chances of security risk as well as physical interference or damage. In addition, remote accessibility of edge devices presents opportunities for data theft and sabotages corporate operations. There are numerous research works available in the literature that address the above-discussed issues. Some of the research publications are survey conclusions related to the security aspects of IoT networks without any specific consideration of EC-assisted network deployments. The contributions made in this survey are listed below:

- We present a summary as well as detailed scrutiny and analysis of security and privacy-related issues about EC-assisted IoT services. Also, security objectives and functions on EC-based IoT applications are discussed.
- A classification of data security threats and attacks due to poor design approaches, misconfigurations, and implementation flaws is discussed. Also, appropriate mitigation techniques for the detection and prevention of attacks are covered.
- Detailed taxonomy of PUF classification based on silicon and non-silicon-based fabrication is presented, and significant performance and quality matrices are discussed.
- A comprehensive summary of AI/ML-based cryptography techniques for the mitigation of data security and privacy threats is presented. Also, the significance of reliable datasets and training data for the development of accurate ML algorithms is discussed in this survey.
- A discussion about future security research goals, privacy-related open challenges, and deeper insights into future research directions in the context of the EC-based IoT ecosystem is offered.

2. Edge Computing

The enormous volume of data generated at IoT sensing nodes can overwhelm any commercial network, bringing all activities on the network to a halt. This leads to increased IT costs, dissatisfied customers causing financial and reputational losses, poor productivity in the industry, and, most importantly, health and safety concerns [87]. EC is the real-time analysis, processing, and storage of data at a location near the source of data where they are generated. Therefore, EC utilizes the available technology that moves computation nearer to the network edge. This involves handling downstream data for cloud services and upstream data for IoT services. [8]. EC brings computational services, data storage, and retrieval as well as diverse enterprise applications close to the actual consumers of information. We can summarize the benefits of EC as it eases the load on the network, cloud, and data center systems while mitigating latency concerns, offers quicker responses, improves application performance and customers' experience. An edge computing platform provides its services by [88]

- processing the sensed data away from the central cloud or data center in real time;
- caching, buffering, and optimization of the data close to edge nodes;
- transforms raw data from edge nodes into a format that can be processed for further deeper analysis.

There are numerous applications and services, such as industrial automation, virtual reality, real-time traffic management, data analytics, and home automation, that leverage the capabilities of EC. These capabilities include features like mobility support, situational awareness, minimal latency, and proximity to edge nodes or users [89]. Edge computing complements cloud computing services through improved user experience in the delay-sensitive application as well as offloading the cloud platform [90]. Although there exist similarities between edge and cloud computing, certain distinct characteristics set them

apart from each other. The location of EC and cloud computing layers in an IoT network is distinctive. Cloud is located significantly from the nodes/users' location and induces high latency compared to EC. Location awareness and mobile support are possible in EC as it is based on a distributed computing model compared to a centralized model of cloud computing [91]. An EC is a subset of cloud computing that comprises hosting diverse services and applications in proximity to sensing nodes and users. As shown in Table 2, there is a significant difference between cloud computing and EC. Also, an edge (location) is different from EC (action). Data collection at the edge (location) and forwarding it to a cloud with limited data processing is not considered to be EC. It is just a case of networking. However, EC occurs if data collection and processing are carried out at the edge of the EC.

Table 2. Comparison of Cloud and Edge computing.

Characteristics	Cloud Computing	Edge Computing
Deployment	Centralized	Distributed
Latency	High	Low
Computational	Unlimited	Limited
Storage	Unlimited	Limited
Scalability	High	Low
Privacy	High risk	Data stays at source
Security	A robust security plan and proactive monitoring against attacks is required	It requires, to a lesser degree, a powerful security plan

2.1. Edge Architecture

Several architectures are proposed for the deployment of the EC layer, but they lack clear definitions and distinctions among nodes. Recent surveys conducted by researchers on EC architectures contain numerous outlooks such as mobile edge cloud servers and networks, application specificity, and considerations regarding resource type, resource management objectives, resource location, and resource utilization. Also, architectural-related challenges like scalability and heterogeneity are elaborated. Premsankar et al. classified all such edge architectures under three categories, i.e., based on the location of resourceful servers from edge devices, resources from heterogeneous edge nodes, and classes of resources at edge and data centers [13].

Figure 3 illustrates a fundamental three-layer architecture for EC. This structure establishes a connection from devices to an edge server, which in turn links to the entire network, encompassing both the cloud and data centers. Within this type of EC architecture, the edge server is situated in a fixed physical location and boasts significant computational capabilities, albeit less powerful than the conventional data centers employed in cloud computing. Furthermore, there is a discernible demarcation between the device level and the edge level, which includes the presence of edge servers [92,93]. The lowest layer includes the IoT sensing nodes responsible for the ingestion of data and applications. It includes IoT devices like cameras, sensors, controllers, industrial machines, etc. The middle layer includes the edge computing infrastructure for data processing, routing, and computing operations. Data generated at the device layer undergo aggregation, analysis, and processing at the edge servers before being transmitted to the upper layer or returned to the device. Although edge computing servers have lower computational ability than cloud servers, they offer better quality of service (QoS) and lower latency than cloud servers. At the topmost layer, there are cloud data centers involving a central data center and interconnected regional data centers. Even in an EC architecture, cloud data centers persist to serve a crucial role as storage places of information. This layer is accountable for tasks such as data analytics, artificial intelligence, machine learning, visualization, and more.

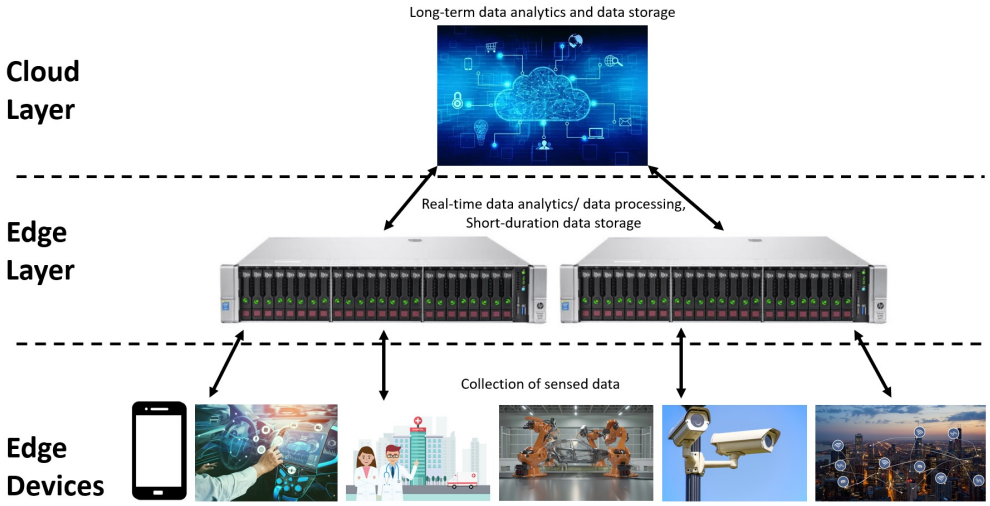

Figure 3. Edge computing architecture.

2.2. Edge Computing Challenges

EC is characterized by higher bandwidth, lower latency, and real-time services, but it is still in the development stages and lacks a well-defined standard framework. As illustrated in Figure 4, the number of edge devices is experiencing rapid growth, creating significant challenges for cloud servers in handling real-time data processing. Statista projects that by 2030, there will be approximately 6.5 billion consumer-focused edge devices, with their average processing speeds advancing exponentially. As a distributed computing technology, EC necessitates well-defined deployment strategies for application workloads on edge nodes. Deployment strategies should be able to answer key questions like where to place a workload, connection policies, and heterogeneity of nodes [94]. EC-driven IoT services create management challenges that organizations should overcome to ensure resilient and reliable operations. Equipment suppliers, service providers, and software vendors are required to work collaboratively to offer cohesive interoperability between various network functions and seamless integration from across edge-to-cloud infrastructure. These factors present challenges in deploying, scaling up, and managing the EC paradigm [95].

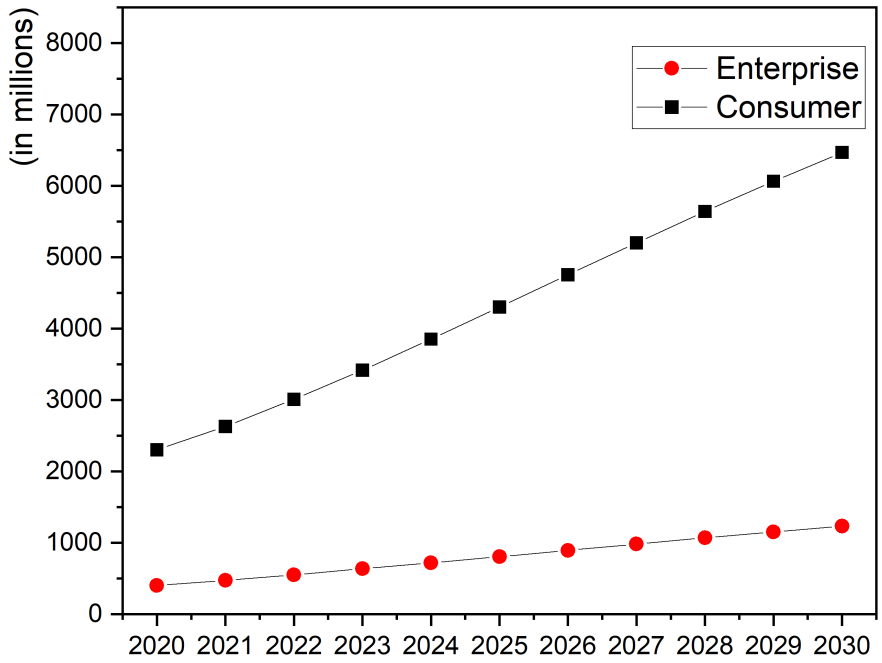

Figure 4. Projected edge devices growth.

Some of the challenges that must be addressed for the widespread adoption of edge computing are discussed below.

- *Heterogeneity.* Many hardware devices and communication standards of diverse natures are deployed at edge networks [96]. EC exhibits heterogeneity across multiple dimensions, including hardware architecture, operating systems, programming languages, accessibility, and the nature of tasks [97]. First, edge devices are diverse, generating data in various formats. Second, data are transmitted through various network access technologies, including 3G, 4G, 5G, WiFi, WiMAX, and LPWAN technologies like Sigfox [98]. Third, the heterogeneous edge nodes providing services encompass a variety of devices such as end-user devices, access points, routers, and switches [49,91].
- *Coordination between communication and computing.* The integration of EC into IoT systems adds significant complexity due to the diverse resource constraints and operational requirements of edge servers and IoT devices [96]. Mobile edge computing (MEC) is a computing model that extends cloud computing to the network's edge [99]. Researchers are exploring the integration of Low Earth Orbit (LEO) satellites with MEC's for low-latency computing offloading services by placing MEC servers on LEO satellites [100] as well as collaborative MECs among connected entities [101]. Network slicing divides a single physical network into multiple virtualized, independent, and tailored networks, aligning with the distributed models of EC. It is managed through the combined optimization of computing and communication resources in EC environments [102].
- *Partitioning and Offloading Tasks.* The computational tasks are divided into smaller sub-tasks, and these tasks are processed either locally on the edge device or offloaded to more powerful edge servers or the cloud. The overall system performance is enhanced by partitioning and offloading tasks while optimally balancing computing and communication resources [103]. Task offloading is a comprehensive process involving application partitioning, decision-making regarding offloading, and executing tasks scattered across the system [104]. The main challenges in designing partitioning and offloading algorithms involve determining the optimal granularity for partitioning, managing resource limitations, adapting to dynamic environments, and addressing the complexity of offloading within blockchain-enabled communication systems [103]. In an MEC system with multiple edge nodes (ENs) serving multiple users, user association is pivotal in shaping the task partitioning strategy, necessitating the joint optimization of task partitioning and user association [105].
- *Security and privacy issues.* EC is vulnerable to access control, identity authentication, information security, and privacy protection-related threats [106]. EC characteristics like geographic distribution, heterogeneity, lower latency, lack of standardized protocols, and operating software expand its attack surface [49,56,96]. Conventional security mechanisms such as attribute or group-signature-based access control, homomorphic encryption, and public-key-based authentication require higher computational ability and storage [107]. Securing edge environments is significantly different from traditional IT security. Implementing security measures on edge devices can potentially hinder their internal operations, impacting the real-time capabilities of edge computing. As a result, a key challenge in edge computing is finding the right balance between minimizing latency and meeting security requirements [108]. Edge operations are typically time-sensitive, safety-critical, and autonomous. The security models implemented in EC networks must accommodate factors such as longer device lifespans and support for legacy infrastructure. Quick patching may not always be possible, particularly if updates require reboots, which could jeopardize safety [109].

- *Monitoring, Accounting, and Billing.* It is important to continuously monitor the usage of EC resources, accounting, and billing-related data for better QoS and charging for EC services. Traditional monitoring and accounting methods typically rely on monitoring interfaces on physical nodes, utilizing hardware probes, and correlating data with control plane and management plane information. However, these approaches often neglect the requirements of the distributed nature intrinsic in an edge environment. A sustainable business model for EC services is needed for monitoring, accounting, and billing purposes. Creating a robust business model proves to be quite challenging due to the mobile nature of users and the limited scope of services. The key focus for EC lies in enhancing resource utilization to its fullest extent and effectively monetizing these resources [110,111].

3. Security and Privacy Challenges

The first level of data processing is at the edge of EC, making them vulnerable to security attacks and data theft associated with end users. Security measures adopted in IoTs include advanced security algorithms like attribute-based access control, authentication based on group signatures, homomorphic encryption, and techniques based on public-key cryptography. Such algorithms demand sizable computational capabilities and memory availability on the devices where they are deployed [107]. The cloud can host almost unlimited resources like memory, computing capabilities, power, etc., but lacks real-time user experience due to its physical distance from IoT end devices. Some research efforts have been made in developing and deploying edge-based security architecture designs like firewalls, Packet filters, intrusion detection systems, side-channel signal analysis, authentication and authorization protocols, privacy-preserving mechanisms, real-time traffic monitoring systems (RTMSs), and cryptographic schemes. Adversaries use various hardware- and software-based techniques to falsify, change, steal, or remove data within edge networks and infect and manipulate edge nodes, devices, or servers found at the edge [112].

Numerous security threats that can compromise user privacy and data integrity or disrupt critical services exist in the edge device layer, communication layer, and edge computing layer in the EC paradigm of IoT network [7,16,113,114]. The commonly identified edge/communication network attacks are eavesdropping, replay attacks, denial-of-service, and jamming [9,115]. The vulnerabilities associated with various edge peripherals within the computing layer are mostly DoS and DDoS attacks [116], whereas ref. [18] has placed DDoS attacks, side-channel attacks, malware injection attacks, and authentication and authorization attacks under the EC infrastructure layer.

However, the research outcomes for edge-based IoT security remain in the early stages of development [107,117]. Initially, EC was assumed to be resilient against cyberattacks since user data no longer needed to travel to cloud servers. Nonetheless, the edge network layer's dynamic nature makes it susceptible to data security threats, as unified security protocols cannot be uniformly applied [118]. Numerous factors contribute to data security and privacy concerns in EC. The vicinity of end users to edge nodes increases the risk of data interception by adversaries. Additionally, the constrained memory and processing capabilities of edge devices, when compared to cloud computing, impede the application of complex encryption techniques and thus aggravate security challenges [119]. It is essential for all stakeholders within an EC ecosystem, like service providers, system and application developers, and end users, to appreciate data security's ethical, legal, and financial implications. Another pressing concern is determining the ownership of sensitive data collected at edge nodes [120]. Mukherjee et al. proposed a layered security framework shown in Figure 5 implemented on cloud EC architecture [121]. Authentication features are imple-

mented at every layer to ensure that only verified end devices can access cloud and edge services. Additionally, location-specific EC is applied at the edge and end-device levels to safeguard user privacy. Moreover, firewalls and intrusion detection systems deployed in both cloud and edge infrastructures help identify and thwart network intrusions. Common security components can exist across multiple layers due to network layer and device requirements, and their purpose and functionality might differ [121]. Robust cryptographic techniques are needed in the cloud edge collaborative architecture, as a huge amount of data flows through unsecured or least secure public channels with a higher probability of privacy leakage and unauthorized data access [122].

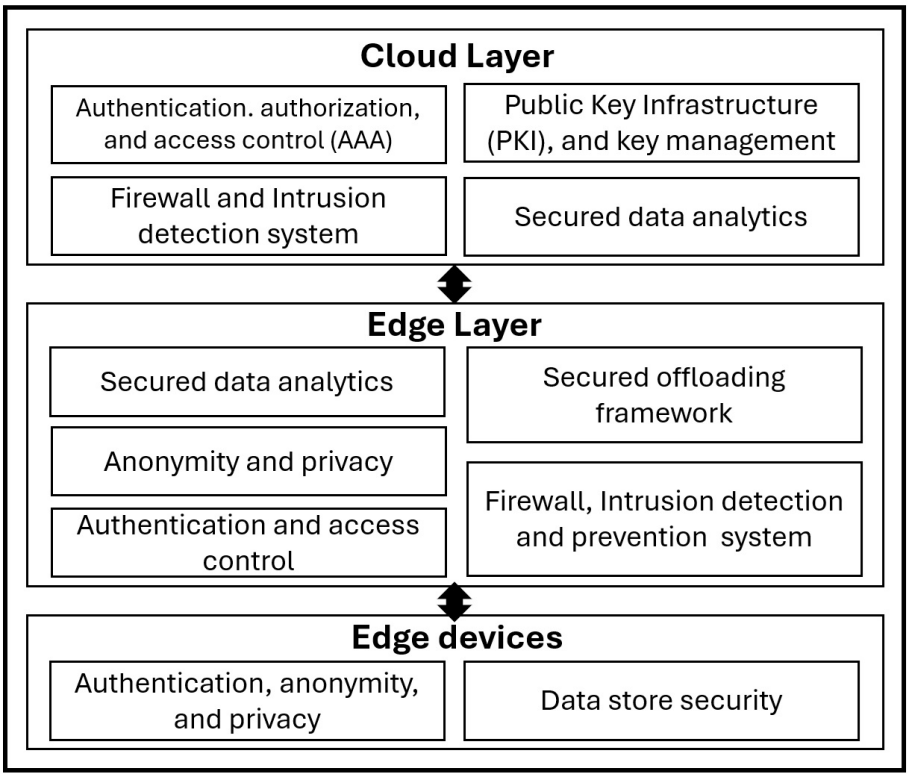

Figure 5. Security functionalities in a cloud-edge computing architecture.

Implementing a uniform security strategy across all edge nodes is extremely difficult due to their management by various users. Wei Yu et al. proposed a problem space of EC-based IoT security defined over three distinct classes, i.e., transmission, storage, and computation [6],

- **Transmission**: Jamming attacks, sniffing attacks, worm propagation, distributed denial-of-service (DDoS), and similar assaults can disrupt data links by choking the network or observing the data flow.
- **Storage**: Innumerable sensors and devices produce a gigantic volume of data, which is then stored across various third-party locations. Such an arrangement poses issues like data integrity being seriously challenged due to the distribution of data into many fractions, resulting in data packet losses as well as data corruption. Also, adversaries can modify or abuse stored data at third-party locations, leading to data leakage and other privacy issues.
- **Computation**: The relocation of computational tasks from the cloud to edge nodes in EC demands an establishment of trust between edge servers and end devices.

3.1. Classification of Edge Computing Security Threats

According to Statista's 2017 report, approximately 159,700 cyberattacks targeted edge networks and were grouped under six distinct groups: side-channel attacks, malware injection attacks, DDoS attacks, man-in-the-middle attacks, authentication and authorization attacks, and corrupt data injection attacks. The percentage share of each class of attacks is shown in Figure 6. User privacy and data security are the most important factors from the service provider's perspective. Sensing network data can extract a lot of private and vulnerable information. For example, access to the data from the electricity and water meters can provide information about the occupancy of a house. There are still open challenges that need to be answered by the EC service providers to protect user-sensitive data.

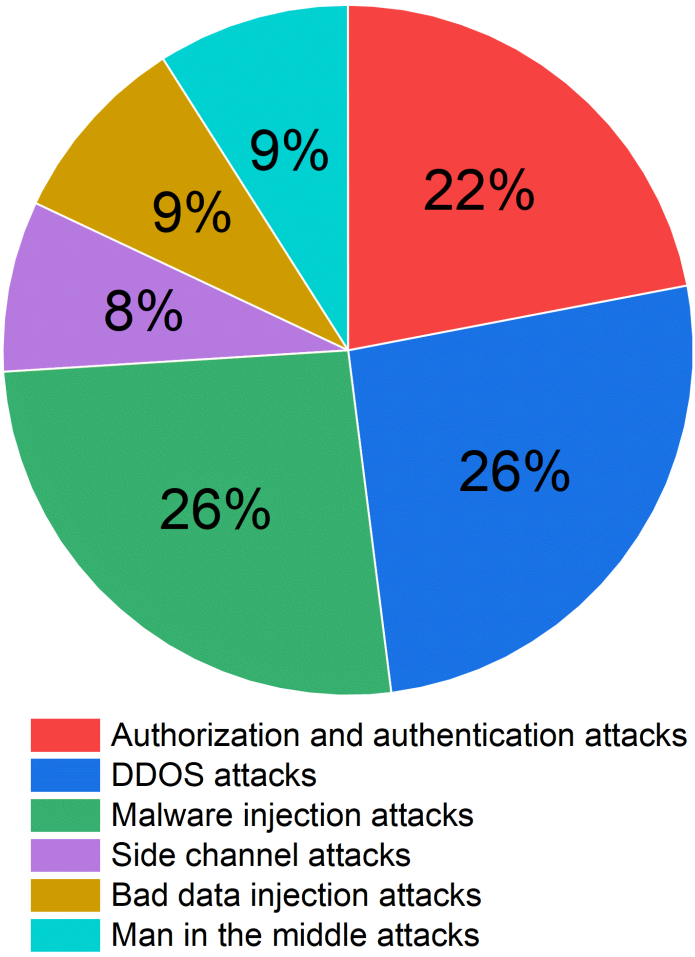

Figure 6. Percentage share of attacks on edge network.

Figure 7 shows a classification of security and privacy threats, listing their types and origins across various levels and layers within EC networks. All stakeholders in EC, including service providers, system and application developers, and end users, must realize their responsibility against data security threats. Another essential data privacy and security issue is establishing the ownership of collected data at the network edge. A suggested solution is to collect and store data at the edge while leaving ownership to the user. Capable tools and technologies are needed to ensure data privacy and security while meeting EC requirements. Edge nodes are resource-constrained, making deploying advanced data security measures difficult due to their resource-intensive nature. Furthermore, the dynamic nature of the location at the network edge increases vulnerability to security attacks and illegal access to user data. Table 3 summarizes the security and privacy challenges as well

as corresponding mitigation techniques against jamming attacks, distributed denial of service (DDoS) attacks, eavesdropping or sniffing, routing information attacks, physical attacks, and privacy leakage.

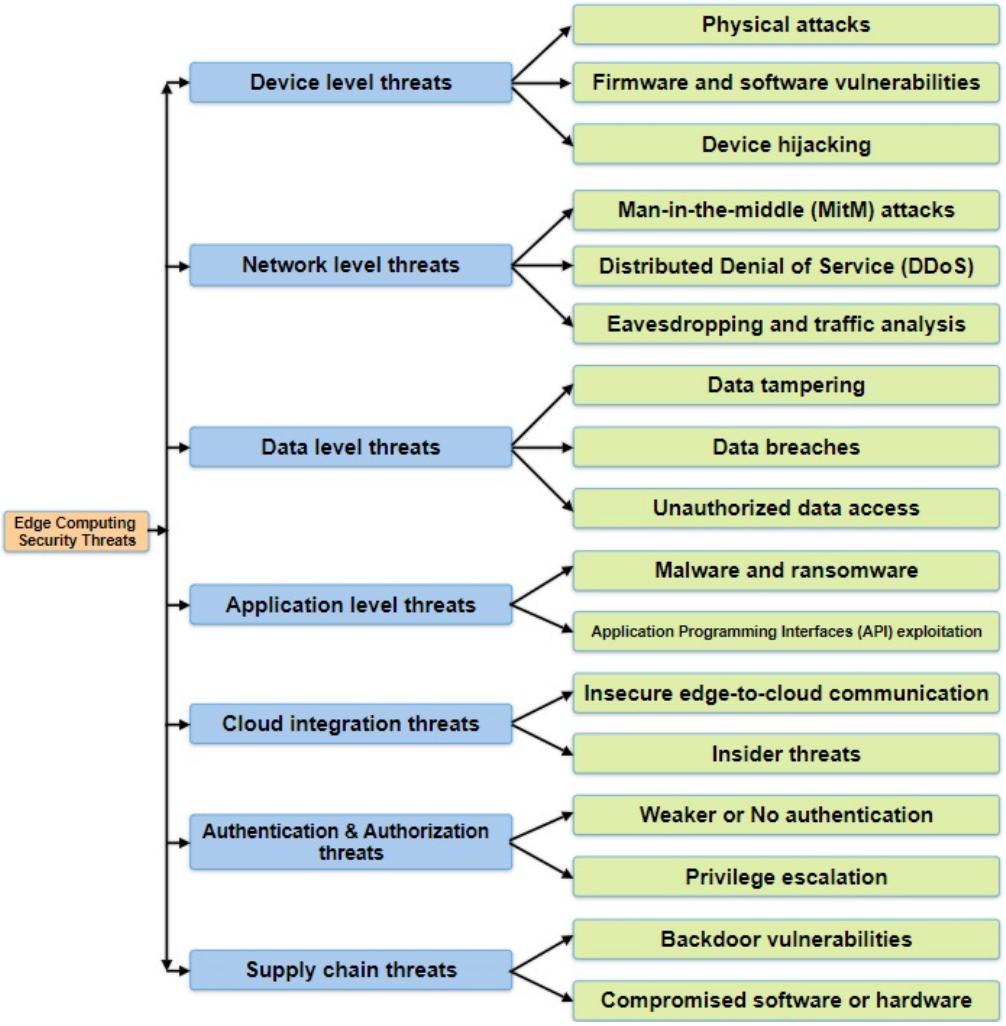

Figure 7. Security threat classification in EC.

Table 3. Classification of edge security threats.

Type of Threat	Description	Mitigation Strategies
Hardware or software malware	Unauthorized hardware or software are injected into the edge network that attack edge servers or devices. Such malware/trojans interrupt network services, and attackers gain control over edge devices and their data.	Side-channel signal analysis, Trojan activation methods, and circuit modification or replacement are the techniques utilized in hardware security [123].
Physical Tampering and Attacks	Attackers may exploit physical access to EC nodes/devices to extract significant and sensitive cryptographic data, manipulate circuits, and alter or corrupt the software and operating systems.	Techniques such as system analysis and self-destruction are employed to inhibit or alleviate the destructive effects of physical alteration and attacks.
Routing Information Attacks	Data throughput, latency, and data paths over a network are affected due to routing attacks. Examples of routing information attacks include black holes, grey holes, wormholes, hello flood, etc.	Monitoring malicious traffic and detecting policy violations can serve as effective countermeasures.

Table 3. *Cont.*

Type of Threat	Description	Mitigation Strategies
Distributed Denial of Service (DDoS) Attacks	The continuous transmission of junk data packets toward the target node can exhaust all resources allocated for handling the malicious data packets. This may result in genuine requests being dropped due to the overload of the target node's resources [56]. Three significant DDoS attacks on edge computing devices are outage attacks, sleep deprivation attacks, and battery-draining attacks.	The Detect-and-filter technique is a tool against flooding attacks. Also, behavior control of devices and policy-based mechanisms within the network can mitigate DDoS attacks.
Privacy Leakage	Privacy leakage in EC mainly involves three separate classes of privacy concerns, i.e., data privacy, location privacy, and identity privacy. Attackers might exploit the location awareness of EC nodes to detect and track device status or to gain access to classified data, posing further risks to privacy.	To address privacy concerns in EC, a privacy-preserving algorithm can be implemented between the cloud server and the edge server or between the end nodes and the edge server [124].
Eavesdropping or Sniffing	An intruder listens over communication channels to gain access to private data, like the physical location of specific nodes, access or control information of the EC node, like node identification or node configuration, message identities (IDs), timestamps, usernames, and passwords.	Data encryption technique at edge nodes with an asymmetric AES algorithm before the transmission on vulnerable channels, the realization of the connection between the edge nodes and edge server, and authentication service between the transmitting and receiving points could overcome eavesdropping attacks [9].
Jamming Attacks	The attacker transmits a wide range of signals with a similar frequency, potentially disrupting network security. Also, unintentional interference is thus triggered in wireless networks due to induced noise and collisions.	The significant transmission parameters like the signal strength of data packets at the physical layer and the packet loss ratio at the application layer serve as indicators of potential jamming attacks [125].
Integrity Attacks Against Machine Learning	ML techniques utilized in EC-assisted Internet of Things (IoT) are susceptible to two different categories of data security attacks. Causative attacks involve manipulating and injecting misleading training datasets to compromise the training process of ML models and Exploratory attacks, where adversaries exploit vulnerabilities.	Researchers propose the use of virtual machines (VMs) with boundaries for running ML processes, hence accelerating the testing and deployment of ML models, and systematic study of attacks in simulated environments or sandboxes [126,127].

Table 4 lists some possible countermeasures against security attacks on edge networks.

Table 4. Mitigating edge network security threats [119].

Strategy	Description
Edge Node Security	Uniform security levels are applied at edge nodes to ensure appropriate safety protocols. Different security levels may allow attackers to break through the nodes with weaker security algorithms.
Full-time Monitoring	Warrants nonstop monitoring of edge nodes while offering network clarity to users through a collaborative interface.

Table 4. *Cont.*

Strategy	Description
Proper Encryption	A complicated algorithm or a secure password exchanged exclusively between legitimate senders and recipients, granting access solely to genuine users.
Intrusion Detection System	Identifies any abnormality or illegal access and alerts the user in case of dubious activities.
User Behavior Profiling	Maintaining a record of users' behavior and keeping track of activities apart from normal behavior to detect an attacker's presence.
Cryptographic Techniques	Secures significant data using codes that block security attacks through a secret key.
Data Confidentiality	Mitigates privacy concerns while restricting unauthorized data transactions, data loss, data tampering, data breaches, and related issues.

3.2. Mitigation Strategies Against EC Security Challenges

The countermeasures against security and privacy challenges in an EC-driven IoT network are discussed in numerous works of literature and can be summarized and placed under classes as shown in Table 5.

- *Cryptographic schemes*: The edge layer, which includes local data centers, as well as sensing devices, is vulnerable to security threats. These edge devices need cybersecurity solutions within limited storage and computation capabilities. A Zero-Trust approach is recommended for securing data in the EC paradigm, with an assumption that all devices have been compromised and all access has to be strictly monitored and authenticated. The standard encryption/decryption methods are memory- and computing-exhaustive [128]. ISO/IEC 29192, Lightweight cryptography is a cryptographic algorithm meant for implementation in constrained environments, including RFID tags, sensors, contactless smart cards, healthcare devices, etc., for the protection of communication protocols.
- *Secured data aggregation, deduplication, analysis*: Data aggregation is a method of clustering the data from various edge nodes by reducing the number of transfers and hence eliminating redundancy. Secure Data Aggregation (SDA) is a highly secure, privacy-preserving, and efficient data compression technique using homomorphic encryption against security attacks such as eavesdropping and forging. The secure deduplication technique removes matching copies of data while supporting data security. It employs Convergent Encryption (CE) for encrypting or decrypting data at the file level, along with a convergent key [129]. Load distribution is used in EC for even distribution of computational, network traffic, data storage, and security-related tasks across edge devices, edge servers, and the cloud. Thus, load distribution prevents edge devices or servers from becoming overwhelmed by diverse tasks while ensuring key security measures like encryption, intrusion detection, and access control are in place. Neto et al. estimated an optimal number of edge nodes that can be assigned to a particular edge server using Equation (1) and further used it in estimating its security factor [130].

$$\Delta_2 = \sum_{i=1}^{n} \frac{\omega_i \times \sum_{\varphi=1}^{\delta} \phi_\varphi \frac{K_{i\varphi} - K_{\varphi_{\min}}}{K_{\varphi_{\max}} - K_{\varphi_{\min}}}}{\sum_{j=1}^{n} \omega_j} \quad (1)$$

$\omega_i (i = 1, \ldots, n)$ represents the number of edge devices associated with a particular edge server. Thus, the percentage of devices assigned to edge server i is found by dividing ω_i by the total number of devices $\left(\sum_{j=1}^{n} \omega_j\right)$. $(\varphi = 1, \ldots, \delta)$ is the min–max normalized security Key Performance Indicator (KPI) while φ regulates priority metrics.

- *Combined with blockchain*: The advantage of implementing blockchain with EC is that it can offer secure data transfer and processing without needing a centralized server by deploying distributed ledger technology. Blockchain governs protocols that collaboratively make judgments involving transaction execution, exercising mechanisms such as voting and consensus algorithms [7]. Blockchain is a distributed and secured ledger technology based on the zero-trust architecture, offering a strong shield against data privacy and security threats [131]. Blockchains are integrated into EC that documents transactions in an increasing chain of blocks [132,133]. As shown in Figure 8, each block is connected to the previous one by referencing its cryptographic hash value, except the first block, the genesis block. Each block contains a significant piece of information like the previous hash, timestamp, counter-like mechanism for every hash estimation called a nonce, Merkle root representing the hash of all the transaction hashes, and transactions (Tx) for a specific time [134]. Consensus algorithms enthuse trust in the network through an agreement among the validated nodes while deciding to generate newer blocks into the blockchain [30].
  Medhane et al. proposed a blockchain-enabled Platform-as-a-Service (PaaS) model that ensures data integrity and security of mobile users in an IoT environment [135]. The behavior detection of blockchain nodes using a technique called T2A2vec is carried out in [136] by extracting node account features, transaction time, transaction type, and transaction amount. The T2A2vec technique counters tampering of transaction records and carries out authentication of blockchain nodes. BeCome is a blockchain-enabled computation offloading measure used in [137] to ensure data integrity in EC. Also, a nondominated sorting genetic algorithm III (NSGA-III), additive weighting (SAW), and multicriteria decision-making (MCDM) are proposed for optimal resource allocation and offloading strategy. Jangirala et al. have adopted a Lightweight Blockchain-enabled RFID-based Authentication Protocol for Supply Chains (LBRAPS) that offers secured and real-time authentication through the integration of blockchain, RFID techniques, and 5G MEC [138]. A decentralized and tamper-proof system using Vickrey–Clarke–Groves (VCG) auction theory is proposed for inducing trust in a collaborative EC while optimizing resource allocation and load balancing [139]. A blockchain-based secured data aggregation (BSDA) approach is used in mobile data collectors (MDCs) for task management and framing of block generation rules [28]. Cheng et al. integrated blockchain, certificateless cryptography, elliptic curve cryptography, and pseudonym-based cryptography methods in a mutual authentication scheme between the edge servers and devices citecheng2021blockchain. Electronic Health Record (EHR) security is ensured by integrating blockchains in EC while storing users' data locally on edge devices [140]. A blockchain user or miner estimates a hash value by solving a computationally intensive proof of work (PoW) linking any two immediate blocks after neighboring miners reach a consensus. However, roadblocks are met in resource-limited nodes of the EC network unable to undertake the mining and consensus process [141].
- *Intrusion Detection System (IDS)*: In EC networks, intrusion detection systems (IDSs) can play a critical role in detecting malicious actions or attacks. IDSs investigate data traffic and resource utilization, issuing alerts when suspicious behavior is detected. IDSs can be characterized into two groups based on their intrusion detection strategies: signature-based and anomaly-based. Signature-based IDSs cross-check monitored

events with a database of known intrusion techniques to identify potential threats. In contrast, anomaly-based IDSs learn the normal activities of the system and report any abnormalities or inconsistent events [71]. Spadaccino et al. and Gyamfi et al. discuss supervised and unsupervised ML models for IDSs for the detection of anomalies in IoT networks and deployment challenges of this ML on constrained edge devices [71,74]. A signature and anomaly-based secured edge computing intrusion detection system (SEC-IDS) framework is proposed in [73] for improved intrusion detection. A hybrid LDA-LR (Linear Discriminant Analysis-Logistic Regression) edge computing model is proposed in [142], utilizing machine learning and an IDS.

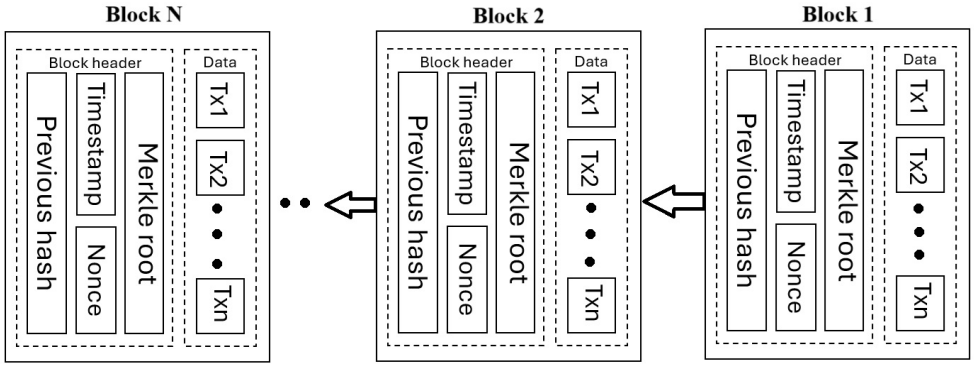

Figure 8. Example of a Blockchain structure.

Table 5. Strategies against EC security threats [96].

Strategy	Network Layer	Limitations
Cryptographic Schemes	Communication Layer	Power efficiency, computational ability, storage, etc.
Secured data aggregation, deduplication, analysis	Data layer	Consumes power, renders sensitive data to intruders' network bandwidth
Combined with Blockchain	Architecture layer	Complex system with more computing capability
Intrusion Detection System (IDS)	Communication Layer	Resource consumption

3.3. AI Role in EC Security

Edge intelligence, or edge AI, represents the blending of machine learning (ML) or artificial intelligence (AI) with EC. Edge AI enables both model training and inference directly at the edge through collaboration between edge devices or utilizing local edge servers near the devices [143]. It is significant for adopting self-learning security solutions at the edges, thus fostering the development of adaptive and autonomous security mechanisms capable of addressing emerging threats in real time [144]. AI algorithms can handle highly unpredictable and complex data while ensuring data security against advanced and evolving threats [145]. Edge intelligence implies a network of interconnected systems and devices conceived for data collection, storage, processing, and analysis near the physical location where the data are generated. This methodology aims to enhance the quality and speed of data processing while improving data privacy and security by preserving sensitive information nearer to its source [146]. The convergence of AI and EC is seen as a natural progression due to their clear intersection. EC is centered around coordinating numerous collaborative edge devices and servers, while AI aims to infuse devices with intelligent behavior by learning from data, thereby simulating human-like intelligence.

AI is important in ensuring data security through its advanced data processing and pattern recognition capabilities [147]. The taxonomy of AI presents numerous techniques like machine learning (ML), Deep Learning (DL), Natural Language Processing (NLP),

Computer Vision (CV), and Robotics [148]. A Venn diagram shown in Figure 9 demonstrates the relation between artificial intelligence (AI), machine learning, deep learning (DL), data science, and data mining techniques [149].

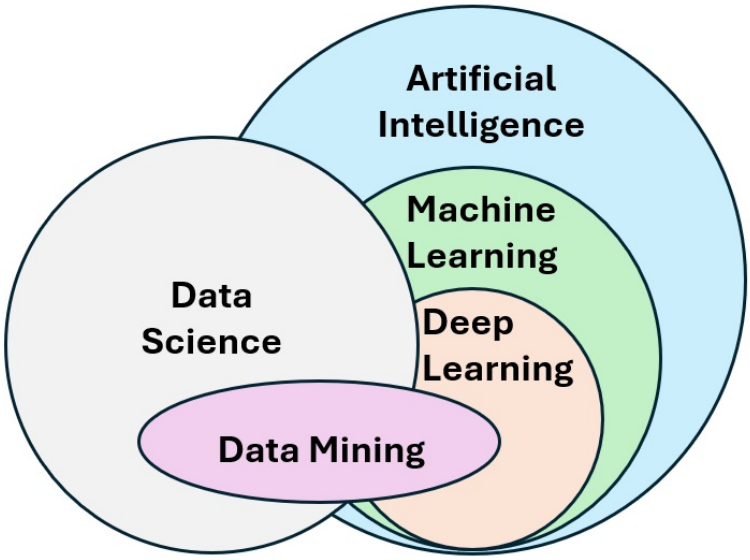

Figure 9. AI taxonomy.

Machine learning (ML), a subset of AI, learns from past data, whereas deep learning (DL), a more specific area within ML, processes data using several nonlinear transformations. DL, compared to traditional ML methods, has demonstrated a remarkable ability to extract and process data, but it also requires sizable computational resources [150]. Decentralized deep learning (DDL), like federated learning (FL) and swarm learning, is a promising tool in securing the data processing at edge devices [151]. ML algorithms for data security can broadly be categorized into transaction algorithms and decision algorithms. Transaction algorithms handle data exploration and preprocessing tasks, while decision algorithms are used to manage business decisions [152]. A major advantage of DL over traditional ML techniques is its ability to automatically extract complex, high-level features from data. DL uses hierarchical neural network models that automatically learn from unstructured data, such as images, sound, text, and video [35]. Wang et al. have discussed numerous techniques which optimize DL models for EC, such as model pruning, quantization, early exit methods, and approaches in DL tasks distribution between cloud and edge nodes [102]. Data science covers various aspects of data processing, including collection, storage, analysis, cleaning, visualization, interpretation, decision-making, value creation, and effectively reporting relevant insights. Data mining aims to uncover newer, hidden patterns and knowledge from data [153].

AI and EC are mutually beneficial to each other as they enable real-time dynamic adjusting and self-optimizing execution of IoT applications. The bottom-to-top arrow shown in Figure 10 represents optimization and the development of EC that requires the assistance of AI algorithms (e.g., computation offloading optimization). Alternatively, the top-to-bottom arrow indicates the need for EC deployment closer to edge devices, hence meeting the latency-sensitive requirements of AI applications [154]. Deng et al. have placed edge intelligence in two groups. The first group, named "AI for Edge", or Intelligence, enables EC and utilizes AI technology in resource allocation, whereas the second group, "AI on Edge", or AI models at the Edge, carries out training of the models and inference at the edge [36]. A hierarchical framework proposed in [155] distributes data fusion and AI processing across three levels, i.e., edge nodes, edge servers, and the cloud.

Data fusion eliminates data redundancy by combining data from multiple sources and thus improving AI performance. The authors of [156] proposed a hybrid learning framework, as current AI-based anomaly detection systems often report false detections. The proposed framework utilizes the Stackelberg game model combined with expert-guided ML rules for higher detection accuracy and minimal false detections. Mitigation techniques against data security and privacy threats are grouped into software-based and hardware-based approaches. In software-based security mechanisms, authentication keys are stored in the non-volatile memories of devices. However, innovations in hardware designs and computational abilities have facilitated data adversaries to breach the security measures adopted under software approaches. Alternatively, hardware-based techniques utilize dedicated hardware-integrated circuits or processors to accomplish cryptographic functions and store access keys. One of the principal challenges with hardware-based security techniques is their susceptibility to man-in-the-middle attacks. In such attacks, hackers can clone the device if the hardware security module becomes compromised. To address these limitations, Gassend et al. proposed hardware-based physically unclonable functions (PUFs) as a security primitive [157]. PUFs leverage intrinsic manufacturing alterations within devices to craft a unique fingerprint of the hardware, rendering it extremely challenging for hackers to reproduce these intrinsic characteristics. However, data acquired from PUFs are vulnerable to environmental factors and the physical conditions of the tested devices. Subsequently, numerous versions of PUFs have been proposed in the literature to enable device identification and authentication, compliant with a tolerable margin of error [158].

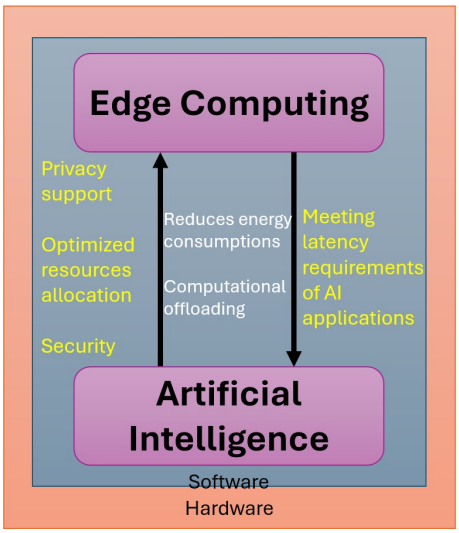

Figure 10. EC and AI: benefitting each other [154].

3.3.1. Machine Learning for Data Security and Privacy

Machine learning (ML) indicates algorithms and statistical models for carrying out specific tasks without explicit instructions. An ML algorithm puts up a mathematical model of user data, also known as a "training set" capable of making predictions or decisions. ML can be used to detect suspicious activity by analyzing user behavior to detect patterns that may indicate malicious activity and ensure data security and privacy requirements [159]. Machine learning (ML) techniques have the potential for enhanced detection of data security and privacy threats while dealing with huge amounts of data coming from IoT end devices. Rigaki et al. mention that the training dataset utilized in the development of ML models is itself vulnerable to a possible data security threat [160]. Usually, the data owners and end users are against the sharing of their sensitive data, which becomes a bottleneck in the development of trusted ML models. To circumvent such issues,

classification protocols utilize ML classifiers over encrypted data to protect the privacy of end users and service providers.

The training approach in centralized ML modeling involves the collection as well as the storage of data in a central location or server. Additionally, in a centralized approach, the intended model is trained using a complete dataset on a central server. This type of approach is practical when the training entity owns the data or has authorization to use it. As shown in Figure 11a, each participant computes their part of the ML model, and subsequently a reduced function finalizes the desired model. However, this technique has disadvantages, including privacy issues due to the distribution of sensitive data with a central cluster of servers and the training process becoming a bottleneck as the dataset grows. Various researchers have proposed an edge-based security system by combining ML with cryptography techniques, which monitors and detects suspicious activities on the network and takes appropriate countermeasures. The deployed ML models include Support Vector Machine (SVM), K-Nearest Neighbor (KNN), and Long Short-Term Memory (LSTM) [161,162].

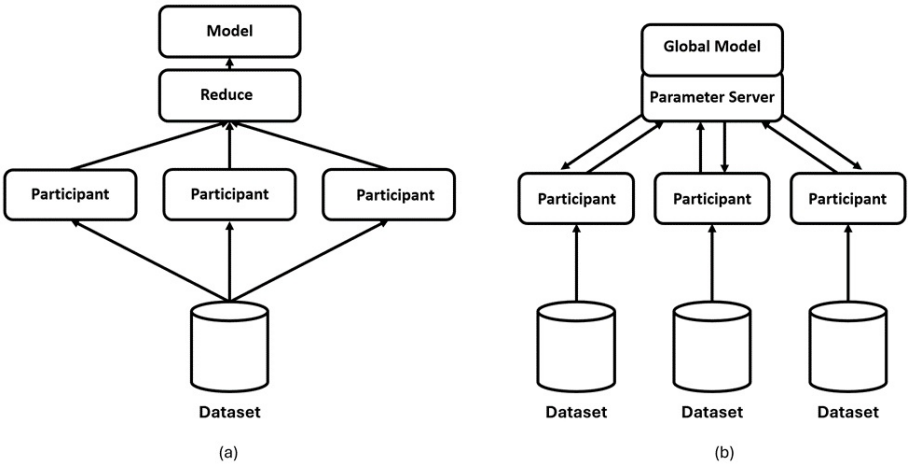

Figure 11. (**a**) Centralized and (**b**) distributed training approaches.

A huge amount of data is required for the training of AI models, and quite often, user-sensitive data become exposed in the process. The integration of AI models with differential privacy ensures the accuracy of models with or without the inclusion of user-sensitive data. The traditional ML models might be robust against data attacks but lack feature extraction from the data and fail to detect attacks that have undergone minor modifications [154]. The study in [163] reviews ML frameworks like TensorFlow Lite, Apache MXNet, and Core ML, along with hardware platforms such as Nvidia Jetson and Google Edge TPU, focusing on their efficiency and accuracy in data processing within edge environments.

3.3.2. Federated Learning

Google has proposed a distributed ML scheme called federated learning (FL) which requires a local ML model at each data site. Later, it trained a complex ML model on an aggregating centralized server [164,165]. FL allows the training of AI models without the need to transmit sensitive data to third-party servers. However, FL networks need a large number of heterogeneous distributed devices, which reduces their communication efficiency. To circumvent the problem of channel efficiency, Feng et al. have proposed a Hierarchical Federated Learning (HFL) framework with an intermediate model aggregator [166]. In a typical distributed learning environment shown in Figure 11b, each participant has access to a local dataset, and the parameter server coordinates the participants. The parameter server in the role of an aggregator has no control over or access to the data stored on

participants. The aggregator server selects participants and aggregates the updated model parameters from the intended participants. Secure model transmission to the server is achieved using cryptographic techniques like Secure Multi-Party Computation (SMC), Differential Privacy (DP), Homomorphic Encryption (HE), etc., among multiple clients without revealing any classified data to each other. Hence, FL has reduced the communication overhead due to the processing of data locally and can offer data privacy and security. Integration of blockchain technology with the FL takes data security to the next higher levels [167]. Blockchain prevents security and privacy threats with its decentralization, immutability, consensus, and transparency characteristics [168]. Among the challenges that FL faces, resource constraints stand at the forefront due to limited power computing nodes and slower communication links. Hence, the FL process at the edge node may take a longer time than expected, as well as energy overheads. Each data source frequently communicates with the central server as the FL model needs to be updated repeatedly and continuously, and there is a higher probability that some nodes upload wrong or old model parameters [169]. A lightweight protocol using secret sharing and a weight masks-based framework is proposed in [170] which protects gradients during FL against attacks like replay and gradient leakage attacks without compromising the model's accuracy.

3.3.3. Multi-Access Edge Computing

Cloud computing capabilities are brought to the edge servers or nodes in a Multi-access Edge Computing (MEC) network shown in Figure 12. MEC exists between the central cloud servers and edge nodes primarily for managing and processing huge amounts of raw data generated from IoT edge devices [171]. It comprises four functional layers, i.e., end devices or hosts, access network, edge network, and core infrastructure. The hosts are connected to the access network, serving as the connection between the functional layers and the Internet. Radio access networks (RANs) establish a connection between the hosts and the remainder of an operator's network. MEC has the potential to improve the quality of service by reducing the end-to-end latency between the edge nodes and data processors, as well as improvement in data security and privacy. MEC also fosters data encryption, authentication, and access control at the edge, thus ensuring authorized access and processing of the data. MEC is deployed either by Mobile Network Operators (MNOs) or by private cloud service providers closer to end customers and has less latency and higher availability [172]. Previous research works focused on resource allocation algorithms rather than ensuring the security of MEC servers and end devices. Of late, limitations of mobile devices and support for resource-intensive applications were introduced by Mobile Cloud Computing (MCC). MCC supports extended battery lifetime, unlimited storage on demand, improved processing capability, and self-service provisioning.

Due to the distributed, small-scale MEC infrastructure, there is less concentration of significant data, thus there is less chance of security and privacy-related attacks. Also, there is a possibility that MEC servers are owned privately, which eases data privacy concerns. For example, the enterprise deployment of MEC skips uploading of users' classified data to remotely located datacentres, as the enterprise administrator manages the authorization, access control, and classifies different levels of service requests at its discretion without involving external parties [173]. MEC can introduce newer classes of services, but its unique characteristics open new types of security and privacy challenges. A huge amount of heterogeneous data generated at IoT edge nodes aggregated, stored, transmitted, and utilized in MEC networks may suffer data leakage incidents [174].

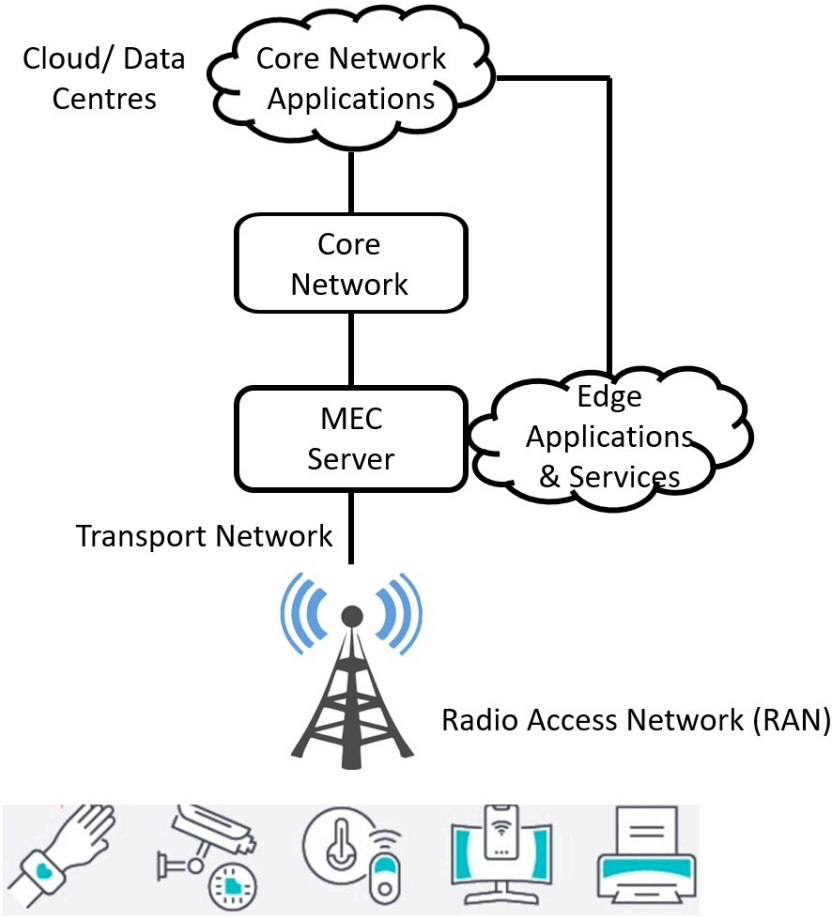

Figure 12. MEC architecture.

3.3.4. Data Anonymization Techniques

Data anonymization is a privacy-preserving technique that masks or removes personally identifiable information (PII) from a dataset to protect the privacy of the users. The user identifiers or PII fall under the direct and indirect identifier types. The attributes that can directly identify a user, such as names, addresses, photos, etc., are direct identifiers, whereas indirect identifiers relate to the attributes that identify users by establishing a relation with other available datasets, like age, salary, occupation, etc. The anonymization techniques have an edge over other privacy-enhancing techniques like encryption, as they do not require key management and large computational resources. However, data anonymization techniques are an irreversible process that provides privacy, but confidentiality or integrity remains unanswered [175]. In recent times, numerous data anonymization techniques have been proposed, including privacy-preserving mechanisms implemented through data masking, pseudonymization, generalization, perturbation, synthetic, etc.

- *Data masking:* Data masking is a technique of concealing data by creating faux versions of sensitive user data by modifying private information. The process involves modification techniques like shuffling, modest word or character substitution, encryption, or masking data. Common types of data masking are static, dynamic, and on-the-fly data masking.
- *Pseudonymization:* Pseudonymization removes user identifiers from the dataset and replaces them with pseudonyms which hides the data source identity. Pseudonymization is defined in the EU-General Data Protection Regulation (GDPR) as "the processing of personal data in such a manner that the personal data can no longer be attributed to a specific data subject without the use of additional information. Such additional

information is kept separately and is subject to technical and organizational measures to ensure that the personal data are not attributed to an identified or identifiable natural person" [176].

- *Generalization:* It is a technique of eliminating identifiable aspects of data to fully remove or reduce its identifiability. Generalization picks up a distinguishable identifier and abstracts it into a more general, lesser distinguishable value. Multiple levels of generalization do exist based on the type of data. An example of a generalization technique is bucketing that groups records into smaller buckets and minimizes the risk of data security challenges [177].
- *Perturbation Methods:* They involve mathematical techniques for the protection of user data privacy. A controlled noise or randomness is added to the data while still being able to perform data analysis. These data privacy techniques are used in various application domains, including ML, statistics, and cryptography. Another method called the differential privacy technique adds a random noise scaled by a privacy parameter to the original data values.

Certain limitations and disadvantages of data anonymization techniques exist, as they reduce the granularity and accuracy of the data. This may damage the relationships between the data points, which is critical for artificial intelligence algorithms or any other data science process. Also, data anonymization techniques can be reverse-engineered by gaining access to external or pseudonym databases.

3.4. Intrusion Detection System

An intrusion detection system (IDS) is a software or hardware-based system, able to detect malicious activity in an IoT network [178]. Also, IDS can track down any violations in the established network protocols or anomalies. Upon threat detection, IDS has two possible responses [179]:

- *Issue alerts:* This class of responses comes from passive IDS systems that issue security alerts via email or text messages. Also, a notification is issued to the security information and event management (SIEM) system, which helps security teams detect user behavior anomalies and apply AI for threat detection and incident response.
- *Countermeasure:* In this class, Active IDS not only sends alerts but also takes countermeasures like changes in access control lists on firewalls to block the suspicious traffic, kill communication-related processes, and redirect traffic to a legitimate part of the network while assessing the threat.

A typical IDS system has three significant units that monitor the network traffic, detect any suspicious activity, and trigger an alert. An IDS can be active, also known as an Intrusion Prevention System (IPS), or passive. An IPS monitors the activities at the system or network level and issues real-time countermeasures in case of threat detection. On the other hand, a passive IDS detects suspicious activity and just alerts the administrators without taking any corrective actions [180]. Traditional IDSs were originally designed for conventional networks but struggled to adapt to the diverse and complex IoT ecosystems. These legacy IDSs proved insufficient in addressing security threats posed by advanced and constantly evolving attacks, such as zero-day exploits. The vast amount of data generated within IoT environments, coupled with highly variable traffic patterns, makes it challenging for IDSs to accurately distinguish between legitimate and malicious activities, increasing the likelihood of errors [181].

In contrast, machine learning (ML)-based IDSs provide more adaptable, scalable, and intelligent solutions to tackle the dynamic nature of IoT security threats. Linear Support Vector Machines (LSVMs), a type of ML algorithm, are commonly used for classification tasks, including intrusion detection, due to their effectiveness in identifying patterns and

anomalies [182]. A classification of IDS based on four main characteristics, i.e., detection method, source of collected data, type of architecture, and response type, is shown in Figure 13. Host-based IDS (HIDS) sits on the host computer and detects malicious behaviors for a single host only. HIDS attempts to detect the presence of unwanted applications in a computing system by analyzing the local data, application registers, log access, and system calls. On the contrary, network-based IDS (NIDS) focuses on detecting malicious patterns in network traffic [183,184]. IDSs normally use one or both of the two primary threat detection methods: signature-based or anomaly-based detection [185].

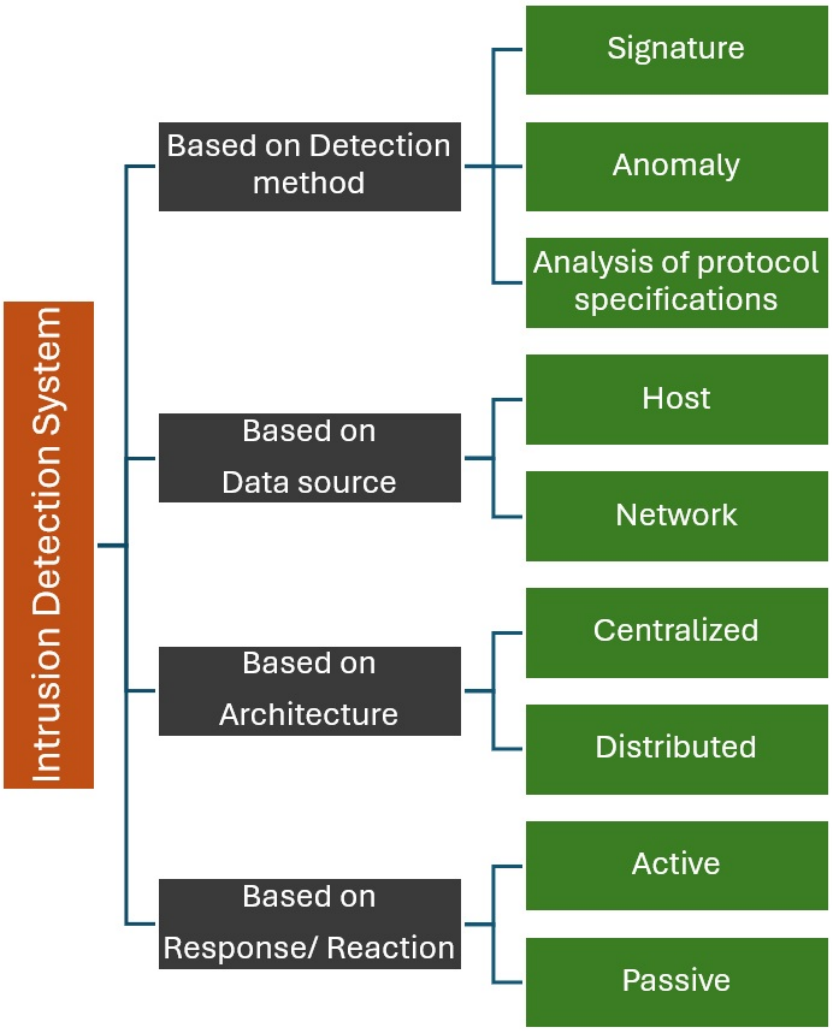

Figure 13. IDS classification.

3.4.1. Signature-Based Intrusion Detection System (SIDS)

SIDS, also known as knowledge-based detection or misuse detection, works on a pattern matching technique to find similar known attacks in the past. An intrusion signature is matched with a database of previously known signatures, and an alarm is raised in the event of a match. In SIDS, host's logs are compared to identify sequences of commands or actions which have previously been identified as malware [178]. Techniques used for generating signatures for SIDS include state machines, formal language string patterns, or semantic conditions. Traditional SIDSs match network packets against a database of signatures and are unable to identify attacks that span over numerous packets. Also, "zero-day" attacks have left SIDS techniques less effective, as there is no prior signature for such attack types. Also, polymorphic malware frequently changes its identifiable characteristics and undermines the adequacy of the SIDS traditional approach [186]. The authors of [187]

reported the detection ability of SIDS against web-based Uniform Resource Identifier (URI) attacks. Three open-source SIDS, i.e., Snort, ModSecurity, and Nemesida, were tested against seven attack datasets using predefined rulesets. The results revealed that untuned SIDSs with the least sensitive configurations were able to detect only 6–8% of attacks, while the most sensitive ones achieved 73–83% with a much lower precision rate of 0.015, thus generating impractical alert volumes.

The researchers have deployed classification models using supervised ML techniques and used a Naive Bayes algorithm-based characterization approach in the probability estimation using network data traffic characteristics. The Naive Bayes algorithm can detect DDoS, DoS, and Code injection attacks on KDD CUP 1999+NSL, UNSW-NB15 datasets. Decision trees are implemented on CICIDS 2017, BOT-IoT, KDDS99, NSL-KDD datasets in identifying attacks such as Sybil, flooding, and spyware threats. SVM utilizes UNSWNB15, KDDCUP99, NSL-KDD, and NOT-IoT datasets in the detection of man-in-the-middle attacks, DoS, DDoS, and tampering. Also, DL techniques like Deep Neural Networks (DNNs), Convolutional Neural Networks (CNNs), and Recurrent Neural Networks (RNNs) are preferred over ML-based approaches while dealing with larger datasets [188,189].

3.4.2. Anomaly-Based Intrusion Detection System (AIDS)

A statistical or knowledge-based ML model is developed in AIDS to detect any significant deviation from the normal behavior, also known as an anomaly. AIDS can be further classified under statistical-based, knowledge-based, and ML-based technique groups based on specific training methods, as shown in Figure 14. A statistical model of normal user behavior is developed from the datasets collected in a statistics-based approach. A knowledge-based method detects desired actions using available system data, such as protocol details and network traffic samples, while an ML approach develops advanced pattern recognition abilities from its training data. There are numerous techniques proposed in the past to model malicious behavior. One of the simplest approaches is based on statistical methods like threshold crossings. However, currently used methods tend to improve traditional detection rates by exploiting the AI capabilities, in particular ML algorithms with an accuracy beyond 95% and much lower false-negative rates [190].

- *Statistical AIDS*: A distribution model of a normal behavior profile is created, and events with lower probabilities are singled out as potential threats. Thus, individual packets are monitored to estimate their statistical metrics, such as the median, mean, mode, and standard deviation, to detect deviations from established normal behavior. A univariate class focuses on a single variable analysis, while multivariate models establish the relationships between two or more variables. In a time series model, the observations are made at set time intervals, and any new or different observation is considered dubious if its probability of occurrence at that given time is too low.
- *Knowledge-based AIDS*: A knowledge base of legitimate traffic profiles is created, and any deviation from the profile is considered an intrusion. This technique is also known as an expert system method that reduces false-positive alarms. However, it needs to update its knowledge regularly due to dynamic computing environments.
- *ML-based AIDS* AIDS exploits ML techniques such as clustering, neural networks, association rules, decision trees, genetic algorithms, and nearest neighbor methods to discover the knowledge from intrusion datasets. The network or host data source and corresponding intrusion or normal as a labelled output value are recorded. A supervised learning method trains a classifier to establish the inherent relationship between the input data and the labelled output value. Fuzzy logic mitigates the high false alarms in IDSs that have numerical data with hard thresholds. The requirement of

labelled datasets does not exist in unsupervised learning environments. There are two different classes of datasets: public and private.

A frequently used public dataset in the past for benchmarking purposes or network security analysis was the DARPA 1998–1999 dataset. An updated version of DARPA 1998–1999 is the Knowledge Discovery and Data Mining (KDD) 1999 dataset, followed by NSL-KDD, and the most recent public datasets are UNSW-NB15 and CICIDS2017 [191]. Yaokumah et al. have conducted an evaluation of Naive Bayes, k-nearest neighbors, decision tree, and random forest ML algorithms on the UNSW-NB 15 dataset for intrusion detection. The experiment results reported an average accuracy of 89.66%, 89.20%, 56.43% and mean absolute error of 0.0252, 0.0242, 0.0867 for random forest, decision tree, and Naive Bayes, respectively. Hence, random forest and decision tree classifiers are a suitable choice for detecting intrusions [192].

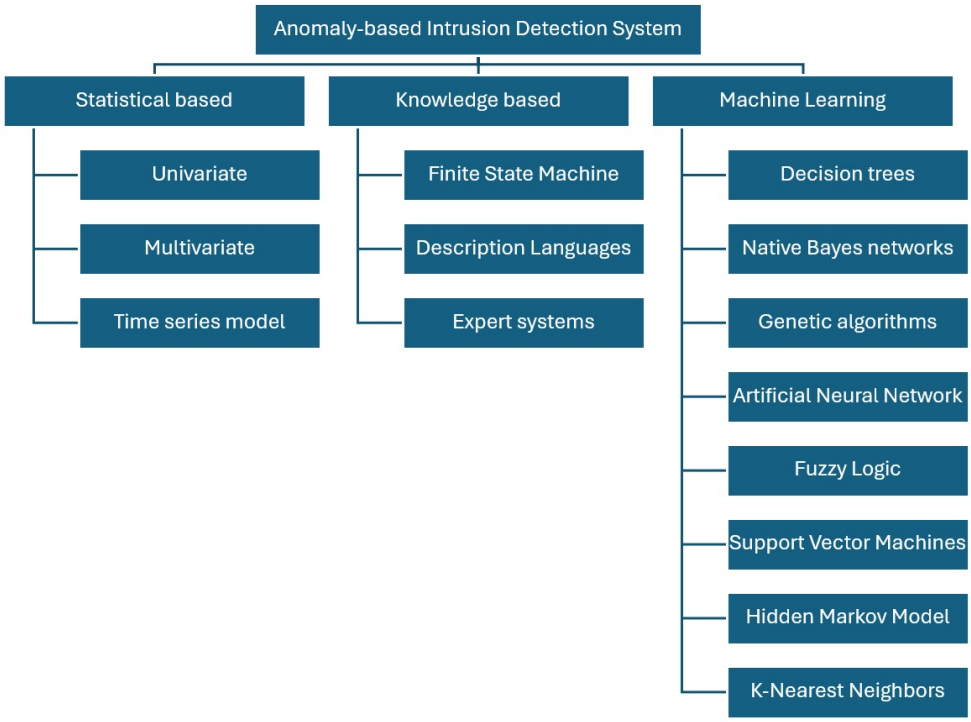

Figure 14. Types of anomaly IDS.

Physical Unclonable Function (PUF) is an alternative authentication scheme without any cryptographic assets burdening the resource-scarce IoT devices.

4. Hardware Security

Edge devices are highly distributed and exposed to numerous threats, including physical tampering, data breaches, and remote cyberattacks. These devices lack standard security practices, deploy heterogeneous communication technologies, and have scalability issues [193]. Thus, strong security measures are required at the hardware level to secure sensitive data and to restrict unauthorized access [194]. Hardware security threats can infiltrate edge devices at any stage of the semiconductor lifecycle, from specification and fabrication to recycling. These threats may arise from unintended design flaws, system side effects, or deliberate malicious modifications during the design process [195]. Both hardware- and software-based mitigation techniques are used to reduce or randomize the vulnerable signal footprints [196]. A widely used authentication technique for edge devices is challenge–response protocols, mostly based on cryptographic primitives and secret keys. However, implementing these protocols on resource-constrained IoT devices remains a

challenge, and the probability of physical threats like direct probing and side-channel attacks is high. Subsequently, a new security primitive, known as PUFs, arrived that offers secure key storage and lightweight authentication [197].

Hardware attacks can be placed into two distinct categories: non-invasive and invasive attacks, based on the level of physical impact on the device [198]. Common hardware security protocols utilize encryption techniques like the Advanced Encryption Standard (AES) and Elliptic Curve Cryptography (ECC) that can be placed under private and public-key encryption. A Hardware Trojan (HT) is a malicious alteration during the chip fabrication stage that might compromise its functionality or spy on encryption keys and forward sensitive chip data to unauthorized devices. HT detection techniques are placed under destructive or non-destructive approaches. Destructive detection includes reverse engineering techniques, such as dismantling IC architecture with Chemical Mechanical Polishing (CMP) and Scanning Electron Microscopy (SEM). However, non-destructive methods analyze IC during the pre-silicon or post-silicon stages. Pre-silicon analysis benchmarks the IC against a fully defined model, while post-silicon analysis includes logic testing and side-channel analysis [199]. The IC supply chain faces security challenges in addition to HT at various stages, including IP piracy, IC cloning, hardware backdoors, and counterfeit chips. On-chip aging sensors can pick counterfeit chips while split manufacturing mitigates IC overproduction and IP piracy issues [200].

Figure 15 lists a broad classification of hardware security threats and corresponding countermeasures available. Reverse engineering (RE) analyzes and decomposes edge devices' design and behavior by extracting confidential data or intellectual property [201]. RE is accomplished by examining various design formats, such as RTL, netlist, layout (GDS-II), mask, or fabricated ICs [195]. It is viable to reverse-trace and refabricate the design, which can be further reused or enhanced [202]. To restrict RE in IC design, hardware obfuscation is the preferred technique that conceals its functionality by placing the logic elements in a random fashion, irregular routing, varying doping concentrations, manipulating dielectric properties, and more [203]. Camouflaging is another option that enables two functional modules to appear identical at the layout level [204]. The adversaries do not physically damage the IoT devices in Side-Channel Attacks (SCAs) nor intervene with or modify the system's operation. SCAs passively monitor specific parameters from sensors or networks, like power consumption, the timing of cryptographic operations, electromagnetic emissions, or acoustic signals [205]. The mitigation techniques against passive SCAs are classified into two groups: hiding and masking. Hiding methods are used for breaking the relation between the processed data and the side-channel leakage, while masking methods disconnect the actual data from the processed data [206]. Counterfeiting is the duplication of hardware devices by cloning or altering the designs without the approval from its creator. It may lead to functional failures in systems and processes but also negatively impact the sales and profits of the businesses involved. The broader consequences of piracy acts extend to public health, safety, and security [207]. The detection of counterfeit devices is difficult as their response against test inputs remain undisputed even in extensive functional testing. However, these counterfeit devices might have hidden malicious characteristics with intentional malfunctions like "back door" for accessing sensitive data [208]. Hardware metering and auditing is a key defense mechanism against hardware counterfeiting, involving tracking of devices. Certain properties of ICs, like negative temperature bias instability (NBTI), hot carrier injection, and electromigration can be monitored by sensors to identify counterfeit or previously used ICs [209]. PUFs are becoming an integral part in security applications, including chip identification and authentication, secure key generation for lightweight encryption, prevention of hardware piracy and counterfeiting, hardware metering, and intellectual property protection [203].

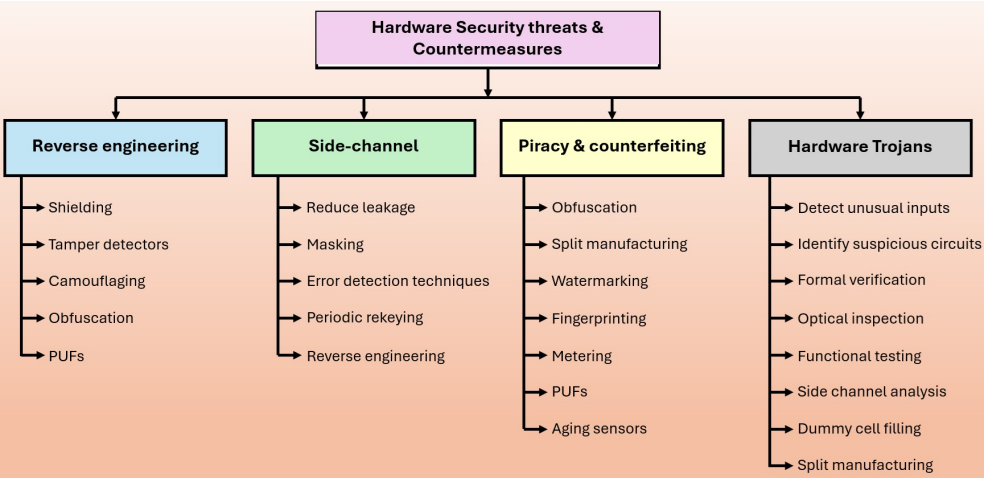

Figure 15. Hardware security threats and countermeasures [210,211].

4.1. Physical Unclonable Functions (PUFs)

Authentication, authorization, and privacy are three essential requirements in an IoT network. Physical Unclonable Functions (PUFs) exploit the inherent randomness created during manufacturing to offer a unique "digital fingerprint" for authentication and secret key storage. Each chip has its fingerprint like those in humans, which is created during the fabrication processes. PUF circuits are triggered by a sequence of input bits known as challenges (C_x) and respond with a sequence of output bits called responses (R_x). No two chips generate identical responses for a common challenge. The combination of an input challenge and its corresponding response is known as a challenge–response pair (CRP) [212]. The process variations during the manufacturing processes of the PUF circuit have a unique silicon fingerprint. Thus, even common input challenges as shown in Figure 16 result in unique challenge–response pairs (CRPs) for the edge devices [213].

PUF carries out an authentication process for an unknown device in two stages, i.e., enrollment and verification. The PUF module receives the challenge bits from the server and the corresponding response bits are stored back into the server by the PUF circuit during the authentication phase. During the verification stage, the server sends the previously stored challenge bits to the IoT device, and the PUF circuit embedded into the device generates response bits. The generated response bits are compared and matched with the CRP look-up table entries for the authentication of the IoT devices. Also, the response bits are used to extract the secret key to ensure confidentiality during data exchanges [214].

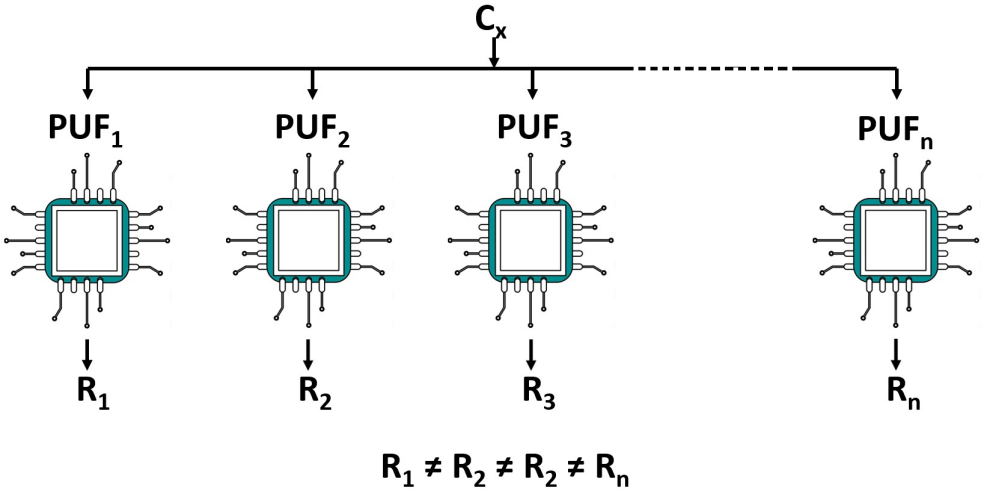

Figure 16. Uniqueness of challenge–response pair (CRP) [215].

PUFs are classified based on their security capabilities, fabrication methodology, physical characteristics, and delay characteristics. Many researchers have presented a taxonomy of PUF under categories like fabrication process and security as illustrated in Figure 17. PUFs are categorized into two types, strong PUFs (SPUFs) and weak PUFs (WPUFs), depending on the number of CRPs. The number of CRPs in SPUFs scales exponentially and linearly in WPUFs with increasing PUF cells. WPUFs are used in storing secret keys or serve as a seed in a random sequence generator [216], while SPUFs can be used for authentication, ID, or key generation [217]. Arbiter PUFs fall under SPUFs, whereas SRAM PUF and butterfly PUF are WPUFs [218]. However, the responses of SPUFs are inherently correlated and highly susceptible to ML attacks, including modeling techniques like Logistic Regression (LR), support vector machines (SVMs), artificial neural networks (ANNs), and ANN-based approximation attacks [219]. The variations in the manufacturing process result in silicon and non-silicon PUF types. The fundamental physical properties of silicon PUFs give rise to three types: analog electronic PUFs, memory-based PUFs, and delay-based PUFs [220]. Non-silicon PUFs create unique characteristics by extracting keys from light beams or lasers, as well as magnetic field strength and radio frequencies, while avoiding the use of electronic signals [221,222].

		Strong	Weak
Silicon PUF	Delay based	Arbiter PUF	RO PUF
		IP PUF	Glitch-PUF
		Clock PUF	PE-PUF
		SC PUF	TERO-PUF
		MC PUF	
	Memory based	BR-PUF	SRAM-PUF
		RRAM PUF	Butterfly-PUF
			SR-Latch-PUF
			Flip-Flop-PUF
			MECCA PUF
			DRAM PUF
	Analogue / Mixed	SNK PUF	ICID-PUF
		ULPC-PUF	C-PUF
		SHIC-PUF	LC-PUF
			PG-PUF
			CN-PUF
			PUF-FSM
			NEM-PUF
			MVL-PUF
Non-Silicon PUF		PH-PUF	Paper-PUF
		Optical-PUF	CD-PUF
		RF-DNA-PUF	Magnetic-PUF
		COPUF	PRNU-PUF

Figure 17. Classification of PUFs.

An arbiter PUF is a delay-based strong PUF that belongs to silicon PUFs. Figure 18 illustrates an N-stage arbiter PUF made up of n pairs of 2-to-1 multiplexers, with each pair in a stage controlled by identical challenge bits. The output, referred to as the "Response", is determined by the differences in path delays. In a standard N-stage arbiter PUF, a rising edge signal travels through one of the 2^N possible paths, guided by the N-bit "Challenge" inputs. An arbiter generates the final response, typically implemented with a D-latch, which decides the output based on the first signal to arrive [223,224]. Optical PUFs have an

edge over other PUF types as they are less noise-sensitive and leverage light diffraction complexity, making them stable and difficult to duplicate [225]. Light acts as the challenge input and generates a unique random pattern as the response [226]. Normally optical structures are not compatible with solid-state integration. However, a recently proposed CMOS imager PUF uses photodiode responsivity under uniform ambient light and dark current variations to generate unique identifiers for camera authentication [227].

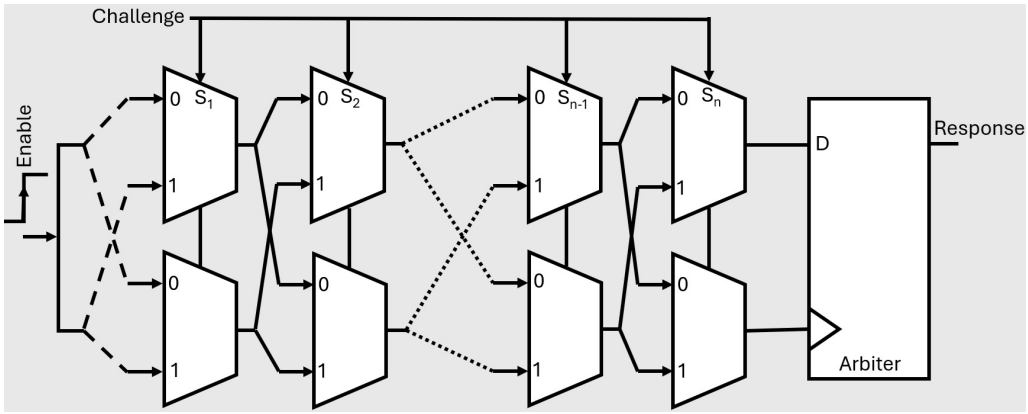

Figure 18. Structure diagram of n-stage APUF.

4.1.1. Strong Versus Weak PUFs

The security and performance characteristics of edge devices in a distributed and uncontrolled environment with limited resources vary significantly with the choice of PUF types. Choosing between weak and strong PUFs in an EC ecosystem depends on numerous factors such as resource requirements, security against threat types, authentication capabilities, reliability, and robustness against physical attacks. Environmental factors like temperature and voltage variations are detrimental to both types of PUFs. The simple and efficient weak PUFs, e.g., SRAM PUFs, are suitable for key generation in secure boot or communication. In contrast, strong PUFs like APUFS are used in devices that require frequent authentication or cryptographic security. Table 6 compares various tradeoff factors of weak and strong PUFs.

Table 6. PUFs trade-offs in the context of EC security.

PUF Types	Merits	Demerits
Weak PUFs	*Low Resource Consumption*: Weak PUFs are simple in implementation, requiring limited hardware resources. For example, SRAM and RO PUFs exploit on-chip memory blocks, thus making them a natural choice for resource-limited edge devices.	Environmental variations such as the ambient temperature or the supply voltage can have a detrimental effect on the performance of weak PUFs [228].
	Low Power Usage: SRAM's minimal static power requirements and quicker access times, or energy-efficient comparison of oscillator frequencies in RO PUFs, make them ideal for edge devices [229].	*Physical Attack Vulnerability*: Weak PUFs are susceptible to probing or cloning attacks; however, exposure of their CRPs is minimal due to their internal operation. Thus, if an attacker gains physical access to the device, they might probe the PUF to recover its response or the keys [230].
	Fast Response Time: The simple architecture and limited set of CRPs of weak CRP suits applications like secure boot due to their quick response, supporting real-time experience in EC.	The responses of weak PUFs are processed internally as a secret key, thus requiring an error correction on-chip and storage of error-correcting helper data. This adds some overhead [231].

Table 6. *Cont.*

PUF Types	Merits	Demerits
Strong PUFs	*Large CRP Space*: A strong PUF generates a large pool of CRPs, hence suitable for authentication tasks, which might lead to frequent authentication between edge devices and servers or peers without repeating challenges [232], thus eliminating the need to store a secret key, as the PUF itself serves as the authentication mechanism.	*Higher Resource Requirements*: Strong PUFs often utilize complex circuits, such as delay lines in APUFs, that require additional area and power needs further straining the limited resources of edge devices.
		Susceptibility to Modeling Attacks: The large number of CRPs from strong PUFs are normally exposed during authentication stages, making them vulnerable to adversaries gaining access to CRPs and able to model the PUF's behavior using ML, compromising its security [233]. Susceptibility to CRP leakage via communication channels or direct interfacing requires additional protection like tamper detection, which may be costly.
		Power and Complexity Trade-Off: A huge number of CRPs consume more energy and require more sophisticated hardware, which may not align with the constraints of low-end edge devices.

4.1.2. Application of PUFs

Physical unclonable functions (PUFs) are used for authentication and secret key storage without needing secure EEPROMs and other expensive hardware. Wang et al. have proposed a Lattice PUF against ML attacks that leverages the Learning With Errors (LWE) cryptographic problem. The designers proposed to build a pseudo-random number generator that integrates a Physically Obfuscated Key (POK) with a LWE decryption function and a linear-feedback shift register (LFSR) [234]. ML capabilities are utilized in the screening of stable challenges to strong PUFs. Initially, randomly generated challenges tested for stability are chosen as the input and output of the ML model for extracting a stable challenge dataset [235]. Wu et al. have proposed a lightweight feedback-based anti-ML-attack Physically Unclonable Function (FLAM-PUF) that integrates an arbiter PUF, a Galois LFSR, and basic logic gates [236]. The design employs a 1-bit feedback mechanism to disrupt the training data, increasing complexity and randomness in the CRP set. This obfuscation reduces the CRP correlation and strengthens resistance to ML attacks by introducing nonlinear relationships. The researchers reported a 50% prediction accuracy against various ML algorithms, including Support Vector Machines (SVMs), Logistic Regression, and Deep Neural Networks (DNNs). A comparable design approach utilizing an LFSR and an Arbiter PUF (APUF) is introduced in [237]. A delay difference quantization strategy for Arbiter PUF (DDQ-APUF) is proposed in [238], which employs multiple configurable delay units (Δ) along two symmetrical signal transmission paths. The design measures and quantifies the delay difference between these two paths. A configurable delay is introduced along the signal path and gradually increases until the output response of the APUF flips. This quantified delay difference is then used as the PUF response, providing robustness against environmental variations. This design follows the Strict Avalanche Criterion (SAC), ensuring that even a minor alteration in the challenge inputs results in significant and random response changes. Wang et al. have proposed a dynamically configured hybrid (DCH) PUF by combining the Self-XOR (SX) PUF with a Modified Feed-Forward (MFF)

PUF. An LFSR is used as a configuration generator, independent of the input challenge. DCH PUF has proven its resilience against diverse ML attacks, including Deep Neural Networks (DNNs), Logistic Regression (LR), and covariance matrix adaptation evolution strategy (CMA-ES) [239]. Zhou et al. have proposed to mitigate ML attacks by reducing linear correlation between the CRPs through a matrix encryption technique called Bagua matrices [240]. This technique is implemented on numerous PUF architectures, including APUF, XOR-APUF, and Multiplexer PUF (MPUF). The prediction accuracy of ML attacks almost reduces to 50% through matrix encryptions, like random guessing, and subsequently improving data security and privacy [241]. The method proposed in [242] combines PUF with Paillier homomorphic encryption or ElGamal encryption to secure data exchanges. Encrypting CRPs during transmission ensures that adversaries cannot intercept or decode sensitive information. Homomorphic encryption enables data verification without decryption, further safeguarding against attacks [242]. A CMOS-based PUF is proposed for device authentication integrated with Elliptic Curve Cryptography (ECC). Elliptic Curve Digital Signature Algorithm (ECDSA) is used in message signing, which enables devices to authenticate themselves without a need for error correction or storage of redundant data [243]. Although APUFs are strong, lightweight, and capable of generating a large number of challenge–response pairs (CRPs), they are susceptible to machine learning (ML) attacks. To counter this vulnerability, researchers in [244] have developed a protocol that authenticates both devices and servers by incorporating an APUF in the device and a PUF model on the server. A zero-transistor interface between the device and server generates "ghost bits" that obscure the challenge bits, making it more difficult for attackers to model the PUF accurately. Another research on cryptography methods for improving strong PUF security and functionality utilizes erasable PUFs, which delete specific challenge-response pairs (CRPs) after their usage [245]. A Configurable Dual State (CDS) PUF, featuring a Feedback Obfuscation Mechanism (FOM), is proposed to enhance hardware efficiency and defend against machine learning-based modeling attacks. The CDS PUF is configured as either a Ring Oscillator (RO) PUF or a Transient Effect Ring Oscillator (TERO) PUF based on the parity of the Hamming weight of the challenge bits. The feedback obfuscation mechanism leverages a stable count value from the RO as a dynamic mask to obscure the input challenge, effectively concealing the relationship between CRPs [246]. A Cyclic Redundancy Check (CRC) PUF alters the seed challenges and transforms the response generation by changing the CRC generator polynomial to mitigate ML-based modeling attacks [247].

A switched-capacitor PUF (SC-PUF) capable of generating stable cryptographic keys leverages metal blocks and capacitive sensing mechanisms. The proposed mechanism protects against invasive physical attacks like focused ion beam (FIB) and probing methods, with a much lower bit error rate (BER) of 10^{-4} [248]. A low-cost resistor–capacitor (RC) PUF is proposed to sense voltage differences caused by the charging and discharging of RC circuits. The experimental results with RC-PUFs have shown 49% uniqueness while achieving over 98% reliability [249]. Cross-PUF attacks exploit power intake measurements from one PUF instance to compromise another, assuming both PUFs originate from the same design file and manufacturing batch. To defend against these attacks, the DRILL method, introduced in [233], integrates Dual-Rail Logic (DRL) with Random Initialization Logic (RIL). This combination reduces the signal-to-noise ratio (SNR) in the power rails and balances power consumption during the transmission of "0" and "1", making it more difficult for attackers to distinguish between the two states. A fuzzy extraction technique is proposed to authenticate biometric data within a lightweight authentication protocol that utilizes blockchains and PUFs [250]. This protocol addresses privacy and security risks, offering protection against threats such as man-in-the-middle attacks, replay attacks,

and impersonation attempts. Similarly, a hybrid approach that integrates blockchain and PUFs is used for device authentication and data integrity that uses PUFs to generate unique device fingerprints [251].

Applications of PUFs include [252,253]

- Identification is an act of claiming identity with a set of attributes, both physical and perceptual, that uniquely define a specific entity. Similar to a biometric identification scheme, PUF response identification can be used to identify the ICs uniquely. A large range of CRPs is stored in the database along with the device ID implemented with the PUF during enrollment. The verifier chooses a CRP from the CRP database. The identification is considered successful if the obtained response and the CRP database output for a specific input are identical.
- Authentication is an act of identity confirmation based on presented attributes. PUFs generate a secure key from intrinsic and inherent entropies created due to variations in the fabricating process. No standard non-volatile storage is needed as randomness is built inside a chip and assures extra protection against the side channel and probing attacks.
- SRAM PUFs, RO PUFs, etc., can generate random numbers with slight modifications in their architecture and find their application in real, or cryptographically secure, random number generators.
- Potential vulnerabilities like copying or reverse engineering can destroy devices' intrinsic and inherent characteristics and thus modify their output. PUFs are suitable for the generation of secrets in cryptography as they are not kept on the hardware and are generated dynamically at device reset.

4.1.3. PUF Performance Indicators

The quality of a PUF is evaluated by metrics like uniqueness, reliability, randomness, correctness, strict avalanche condition (SAC), etc., that verify its applicability to a specific application. PUF metrics are measured by collecting response bits against a set of challenges to the PUF. A specific application has unique sets of requirements; hence, all metrics are not equally important [254,255].

Uniqueness: It is a PUF characteristic representing its ability to generate a unique response against a similar set of challenges subjected to each die in a lot [256]. Uniqueness is the average inter-chip Hamming Distance (HD) of the responses collected from a group of chips. The uniqueness value of an ideal PUF is about 50%, meaning half of the bits in the responses of the PUFs should be different [254]. For example, in an FPGA-based k n-bit, PUF responses are P_1, P_2, $\cdots$, P_k, then the average Hamming distance given by Equation (2), is the measure of uniqueness [78],

$$u = \frac{2}{k(k-1)} \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \frac{HD(P_i, P_j)}{n} \times 100\% \tag{2}$$

Reliability: The PUF and CRP under noisy and variable environmental conditions are measured by their reliability, i.e., the PUF outputs the same response under variable operating conditions. However, numerous environmental conditions like temperature, voltage, and aging of the devices are responsible for variations in the PUF signatures. The ideal value for reliability is 100% and it can be estimated using Equation (3).

$$u = \frac{1}{x} \sum_{y=1}^{x} \frac{HD(R_i, R_i, y)}{n} \times 100\% \tag{3}$$

where x represents the times of sampling; n is the number of bits of a signature generated by a PUF; $R_{i,y}$ is the yth sampling of R_i.

Randomness: It is a measure of the PUF's ability to generate 0 or 1 in its response bits with equal probabilities. The randomness of a PUF should be 100% in an ideal case. PUF-based authentication protocols rely heavily on random physical imperfections that occur during the semiconductor manufacturing process, thus creating static randomness. However, the identification (ID) extraction from the PUF becomes corrupted due to dynamic randomness sources like noise which reduces the PUF's reliability [257].

$$Randomness = 1 - |P_r(ID = 0) - P_r(ID = 1)| \tag{4}$$

For 2^M challenges, the probabilities to obtain an ID at 0 and 1 can be given as

$$Randomness = 1 - \left|\text{erf}\left(\frac{E(D_R)}{\sigma\sqrt{2 \cdot M}}\right)\right| \tag{5}$$

where D_R is the pdf of

$$\sum_{i=1}^{M} d_{c_i}^i \tag{6}$$

For a variance of $M\sigma^2$, the randomness expression Equation (7) is given by [258]

$$\begin{aligned} P_R(ID = 0) &= 1 - P_R(ID = 1) \\ &= P_r\left(\textstyle\sum_{i=1}^{M} d_{c_i}^i < 0\right) \end{aligned} \tag{7}$$

Correctness, Bit Aliasing, Uniformity, and Steadiness are additional PUF performance metrics discussed in the literature [78,254,256].

Table 7 presents a comparison of PUF performance metrics mentioned in the previous section. It is inferred from the table that the Uniqueness and Uniformity performance metrics of Lattice PUF remain closer to ideal values whereas RC-PUF is the lowest-performing one.

Table 7. Comparing the performance of PUFs.

Article	PUF Type	Stages	Uniqueness [mean (std)]	Uniformity [mean (std)]	Reliability [mean (std)]
[234]	Lattice PUF	1000	50.00% (1.58%)	49.98% (1.58%)	1.26% (2.88%)
[236]	FLAM-PUF	64/128	49.73%/49.99%	49.81%/49.85%	95.59%/96.58%
[237]	Strong response–feedback PUF	32/64/128	50.17 (1.41)/50.00 (0.31)/ 49.99 (0.21)	49.54 (3.67)/50.05 (2.79)/ 49.93 (1.78)	-
[238]	DDQ-APUF	64/128	47.28%/47.65%	50%/50%	99.95%/99.91%
[246]	FOM-CDS PUF	17	47.38%(RO mode)/ 53.79% (TERO mode)/ 50.33% (Full mode)	47.71% (RO mode)/ 56.23% (TERO mode)/ 53.68% (Full mode)	3.1% CRO-PUF/ 9.14% Dual mode/ 7.91% FOM-CDS PUF
[247]	CRC-PUF	128	49.9978%	50.0777%	-
[249]	RC-PUF	32	27.3% (bit delay = 2 µs)/ 30.9% (bit delay = 32 µs)	50.3% (bit delay = 2 µs)/ 50.3% (bit delay = 32 µs)	96.2% (bit delay = 2 µs)/ 98.5% (bit delay = 32 µs)

4.1.4. PUFs as a Root of Trust

A layered defense model, as shown in Figure 19, is preferred for a secure system with outermost layers managing the regular operations of the device and acting as protection barriers for inner layers. RoTs act as a fundamental source for various secure schemes enforcing access to cryptographic modules as well as security resources at the hardware level.

The software security built on top of hardware-based RoT provides extra layers of flexibility and protection. These hardware-based RoTs build a trusted execution environment (TEE) for running privileged software, perform cryptographic operations, and offering constant tamper protection. This design approach minimizes the attack surface area and makes inner layers easier to secure because they have fewer, highly controlled tasks. The trust–validation sequence continues moving towards inner layers up to the system core, known as the Root of Trust (RoT) [259]. Edge devices leverage RoTs in establishing a protected environment for cryptographic processes needed for data encryption and authenticating devices connected to backend systems [260]. RoT applies various code validation mechanisms before executing the code on secured CPUs and shields against physical attacks to a certain extent. Thus, a Chain of Trust is established when each component in this chain trusts the codes it runs as they are validated by the previous link, creating an unbroken line of trust back to the Root of Trust [261,262]. The hardware RoT secures EC operations by providing the cryptographic keys in the booting process. Hardware-based RoT is typically a small, dedicated chip embedded within an IoT device leveraging upon intrinsic hardware characteristics [263]. PUFs are ideal for hardware-based RoT that hosts cryptographic functions, such as private and public key encryption [264]. The unique keys generated from the edge device's PUF and the secure boot process ensure that only authorized firmware or updates are loaded, preventing trojan or malware attacks.

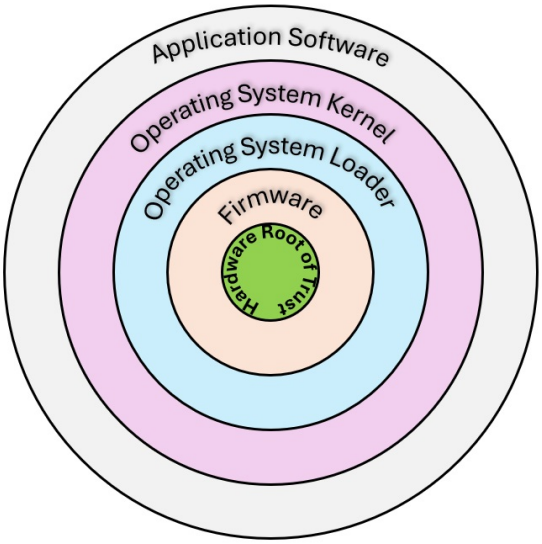

Figure 19. Layered defense model.

Rojas et al. proposed a hardware Root of Trust (RoT) architecture utilizing a Zynq-7000 SoC FPGA (Xilinx Inc., San Jose, CA, USA) and integrating various cryptographic components. These components include PUFs for device authentication, the Advanced Encryption Standard (AES) for data encryption, Secure Hash Algorithms (SHA-2 and SHA-3) for ensuring data integrity, and the Edwards-curve Digital Signature Algorithm (EdDSA) for digital signature verification [265]. A hardware RoT is proposed in [266], leveraging Quantum Tunneling PUFs to identify ICs digitally. In contrast to SRAM PUFs, Quantum Tunneling PUFs operate without the need for error correction. The software-based PUF (SW-PUF) combines physical chip variations with delays in software instructions to generate unique IDs within a secure Root of Trust (RoT). This approach supports secure boot and remote attestation, ensuring that only authenticated, tamper-free software is executed [267]. A secured IoT architecture proposed in [268] combines PUF with Trusted Platform Module (TPM), and Tangle Distributed Ledger Technology (DLT acts as a RoT, establishing a unique digital identity for each device. The proposed architecture implements a Security-by-Design

(SbD) approach at the hardware level, strengthens attack resistance, and defends device and data integrity. Quantum channels are vulnerable to diverse noise sources, which include environmental interactions and eavesdropping attempts. A key reconciliation protocol is proposed in [269], allowing transmission of a bit stream through insecure and noisy quantum channels. Also, the researchers claim that the proposed protocol can reconcile two PUF responses obtained from the same challenge but at a different time. Also, minor noise levels in the PUF responses are mitigated through the application of a fuzzy extractor, designed to produce stable cryptographic keys from marginally erratic PUF responses [270].

4.1.5. Integration of FPGAs-Based PUFs with Edge AI

Artificial Intelligence (AI) assisted data analytics at the edge, allowing for improved interpretation of raw and unstructured data from the physical world. AI at the edge has the potential to automate complex and advanced tasks while preventing user-sensitive data from being transmitted over the network and into data centers at the same time. Edge AI models human reasoning, thus enabling machines to sense, comprehend, perform intelligent detection, and transmit results to the cloud for long-term storage or big data processing. It is capable of recognizing and fighting back against cyberattacks as well as other cyber threats based on the continuous input of data, identifying patterns, and back-tracking the attacks. Data privacy and security breaches need to be taken seriously as they may cause business interruptions, revenue losses, and panic among the public [271]. The human brain comprises nearly 100 billion neurons, and over 100 trillion connections are established to form a network of neurons which in turn significantly influences the brain's capabilities. The interconnectivity within an FPGA resembles the neural wiring of the human brain, and its programmable logic fabric offers the flexibility of the brain [272].

The dynamically reconfigurable as well as customizable hardware architecture of Field Programmable Gate Arrays (FPGAs) has offered a promising solution in accelerating compute-intensive workloads [37]. FPGA-based edge network accelerators offload intelligence, data processing, analytics, and communication capabilities from the cloud to where the data originates [273]. Cloud computing provides the infrastructure needed for securing users' data as well as maintaining their integrity and privacy. However, there is no foolproof technique yet that guarantees data protection nor a processor that can isolate the execution of users' applications from data theft. FPGAs are capable of providing stronger security guarantees as there is no need to involve vulnerable operating systems, drivers, or compilers, nor any other system software [274].

The possibility of incorporating general-purpose processors such as soft cores on FPGAs makes these reconfigurable devices suitable for IoT applications as they can provide solutions with enhanced security, reduced size, energy consumption, and cost [275]. Silicon chip fabricators and designers have integrated FPGA and ARM processor cores for efficient edge AI processing. Also, the benefits of shorter development time make an FPGA-based solution the ideal choice for an intelligent edge device [276]. Integrated chip manufacturers mostly outsource their operations, where intellectual property (IP) theft poses serious concerns. In contrast, FPGA designers do not configure them with sensitive IPs unless the delivery of the product is completed [277]. Cybercriminals can replicate FPGA applications by intercepting their programming bitstream or reading the internal memory. Modern FPGAs have started using advanced encryption key standard (AES) with the battery-backed SRAM 256-bit or 384-bit security key, AES with the eFUSE key, on-chip bitstream keyed-Hash Message Authentication Code (HMAC) algorithm, bitstream authentication, etc., can mitigate the risks, protect intellectual property, and improve the overall safety of FPGA devices.

FPGA-based edge devices exploit AI and ML capabilities for the processing of sensed data and subsequently reduce network bandwidth requirements and dependence on cloud processing. Also, vendors are providing IP cores like OpenVINO, Vitis-AI, etc., to leverage FPGA interfaces for the optimization and deployment of deep learning (DL) models [278]. Open Visual Inference and Neural Network Optimization (OpenVINO) is an open-source toolkit from Intel that facilitates quicker inference of deep learning models on hardware accelerators and easy heterogeneous execution across numerous hardware platforms. Deployment of the OpenVINO toolkit and the Intel FPGA AI Suite in the development of DL-enhanced embedded systems on multiple FPGA-accelerated servers is shown in Figure 20. The OpenVINO toolkit comprises tools and libraries that utilize techniques like pruning, quantization, etc., for the optimization of neural networks. The basic workflow of Intel Distribution of the OpenVINO toolkit is as follows:

- **Model Optimizer** converts models from various frameworks like Caffe, TensorFlow, Open Neural Network Exchange (ONNX), and Kaldi to an intermediate representation format for faster inference.
- **Inference Engine** reads the IR format and supports heterogeneous execution across different hardware architectures such as CPU, GPU, Integrated GPU, etc.
- **Model Zoo** is a common interface for heterogeneous hardware that contains examples to get started with OpenVINO quickly.

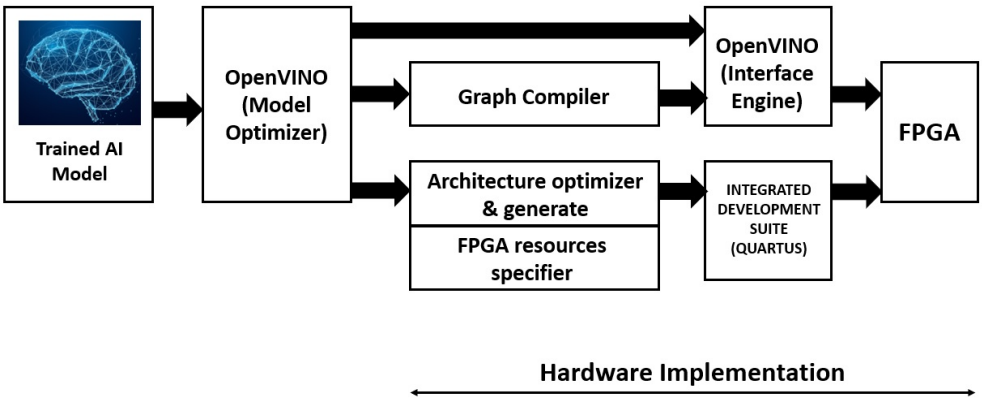

Figure 20. Edge-ready AI toolkits for Intel FPGAs [272].

Vitis AI 3.0 (Xilinx Inc., San Jose, CA, USA), is a unified software platform that includes optimized IP, tooling, and libraries to grant users access to AI inference acceleration through adaptable hardware. It consists of a rich set of AI models, optimized deep learning processor unit (DPU) cores, tools, libraries, and example designs for AI at the edge and in the data center. It provides a unified programming model for accelerating Edge, Cloud, and Hybrid computing applications. Vitis AI integrated development environment is presented in Figure 21, with the target platform, i.e., FPGAs, as the base layer. The Xilinx runtime library in the second layer controls the data movement across domains. Also, compilers are used in the layer for mapping the AI model's optimal instruction set and dataflow model as well as carrying out optimization tasks. There are more than 400 optimized and open-source applications across eight Vitis libraries that are defined in the third layer and offer out-of-the-box acceleration with minimal to zero code changes to your existing applications [279].

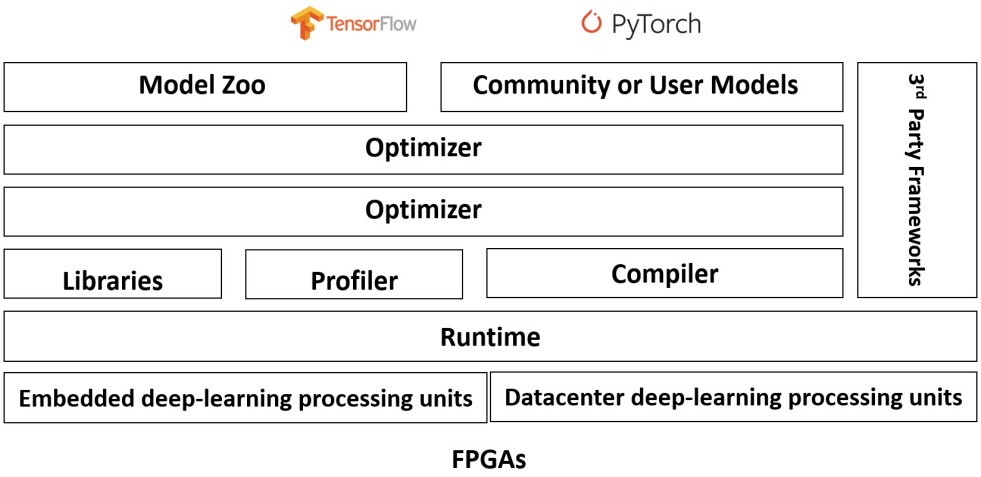

Figure 21. Xilinx *Vitis*^{TM} AI integrated development environment.

5. Open Research Issues

The motivation of this section is to introduce research open challenges and opportunities in the security and privacy issues related to the EC paradigm. The centralized computational approach in data centers and hyperscale clouds is robust against security threats as it "hides" the user's data behind layers of security defenses, both virtually and physically. However, EC faces many security challenges, and here we present some of the open research challenges as well as the scope for further work [96].

- **Heterogeneous EC architecture**: The users in a traditional cloud computing approach are masked from the hardware in place and how software/application performance depends on hardware resources. EC introduces complexity and a need for multi-layered security schemes because of an assortment of standards and protocols [280]. It introduces the need for unique data propagation management schemes among the heterogeneous edge devices [121]. Data privacy is achievable through encryption techniques, but EC architecture makes the existing encryption schemes too cumbersome for the limited processing resources [281]. Furthermore, research needs to focus on ML algorithms explicitly for IoT forensics, matching the distinct features of diverse IoT devices. The potential of emerging technologies such as blockchain needs to be explored in securing digital networks [282,283].
- **Dynamic resources allocation**: Contrarily to cloud computing, the resources in the EC network are rather limited; thus, static allocation techniques cannot achieve optimal resource utilization. The dynamic allocation of computing and storage resources in a distributed EC network remains a bigger challenge. The resource allocation strategy in EC is important for ensuring efficient and effective use of resources and maintaining the quality of service (QoS) for applications that demand real-time data processing and low-latency response. The task of partitioning in EC poses the challenge of optimal partitioning and faces challenges in dynamic resource allocation without the computational or storage capacity or location of edge nodes. Several obstacles exist in the deployment and optimization of resource scheduling in EC and cloud collaboration using deep reinforcement learning (DRL). A significant challenge remains in terms of higher computational cost and extended convergence times of DRL algorithms [284].
- **Data abstraction**: The edge node needs a certain amount of training data to carry out analysis tasks. Data abstraction carries out data preprocessing techniques like noise cancellation, data classification, and data computation. Heterogeneous devices use different data formats, and data security algorithms cannot be fed with a complete set of raw data, but it should only abstract the relevant part. Storage is a limiting

factor while selecting the size of raw data and prediction accuracy. The selection of an optimal data abstraction technique is not easy because of the heterogeneity of devices, different data formats, and different corresponding operations. Thus, a unified architecture or interface standard for the EC-based IoT applications that supports migration between diverse embedded operating systems is needed for the abstraction of IoT and edge devices [96].

- **Secured EC nodes**: Devices in an EC network need a foolproof access control and an end-to-end threat protection mechanism. Edge security refers to device security, network security, data security, and application-level security, focused mainly on the protection and privacy of user data. Mitigation strategies include first the risk definition, uncompromised device functionality, multiuser edge node security, and minimal service levels at user nodes. The development of authentication mechanisms for specific edge nodes and the privacy module is needed to maintain the trustworthiness of edge data centers [285].
- **Federated learning (FL)**: FL refers to a secured ML technique in a distributed environment comprising scattered edge devices or servers while ensuring the user data do not leave the source premises [286]. The research for fullproof privacy and attack mitigation techniques remains a focus of FL. In addition to data security challenges, the communication overhead of FL is comparable to the computational overhead. The two significant attacks against FL are poisoning attacks and byzantine attacks. The poisoning attack includes the act of tampering, destroying, or corrupting the edge data used in local training or model generation [287]. Poisoning attacks are relevant to a single edge node or a server, while byzantine attacks are prevalent in the collusion of multi-users distributed learning environment [288].

6. Conclusions

Our current research thoroughly examined and summarized the challenges related to data security and privacy preservation in EC, along with corresponding countermeasures. We also discussed the advantages and limitations of integrated EC and IoT paradigms. Furthermore, we performed an in-depth analysis of security and privacy issues within EC-assisted IoT networks, including a comprehensive survey of potential security attacks and their countermeasures. We researched how state-of-the-art technologies, including PUFs, AI, IoT, and ML, can mitigate security-related challenges in an EC paradigm. Given that resource-limited edge devices may not support traditional cryptographic security solutions, lightweight security primitives like PUFs are an alternative solution for low-cost key generation. Additionally, we conducted a detailed examination of AI/ ML-based security mechanisms, categorizing them extensively. We also provided insights into commercially available toolkits from leading manufacturers and developers utilized in deploying EC services. Finally, we identified open research directions and gaps in data security and privacy issues within EC, outlining areas for future investigation and development.

Author Contributions: Conceptualization, A.M.S., M.R.I. and M.H.H.; methodology, A.M.S., M.H.H., M.R.I. and A.K.; formal analysis, A.M.S., S.A.Z. and M.H.H.; investigation, A.M.S., S.A.Z. and A.R.B.N.; resources, A.M.S., M.R.I. and A.K.; data curation, A.M.S. and M.H.H.; writing—original draft preparation, A.M.S. and M.H.H.; writing—review and editing, A.M.S., M.R.I., M.H.H., A.K., S.A.Z. and A.R.B.N.; visualization, A.M.S. and M.H.H.; supervision, M.R.I. and M.H.H. and A.K.; project administration, A.M.S.; funding acquisition, A.M.S. and A.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work is part of the A'Sharqiyah University, Oman-Internal Research Grant (IRG-16), 2024-26 "Intrusion detection in an IoT network through Machine Learning (ML) of hardware characteristics".

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding authors.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AES	Advanced Encryption Key Standard
AI	Artificial Intelligence
AMQP	Advanced Message Queuing Protocol
BR-PUF	Bistable Ring PUF
CN-PUF	Carbon Nanotube-Based PUF
CO-PUF	Computational Optical PUF
CRP	Challenge Response Pair
DDoS	Distributed Denial-of-Service
DL	Deep Learning
DP	Differential Privacy
DPU	Deep Learning Processor Unit
EC	Edge Computing
FL	Federated Learning
FPGAs	Field Programmable Gate Arrays
GDPR	General Data Protection Regulation
HD	Hamming Distance
HE	Homomorphic Encryption
HMAC	Hash Message Authentication Code
ICN	Information-Centric Networking
IDS	Intrusion Detection System
MCC	Mobile Cloud Computing
MEC	Multi-Access Edge Computing
MECCA-PUF	Memory Cell-Based Chip Authentication PUF
ML	Machine Learning
MNO	Mobile Network Operators
MQTT	Message Queue Telemetry Transport
MVL-PUF	Multiple-Valued Logic PUF
NEM-PUF	Nano-Eectromechanical PUF
ONNX	Open Neural Network Exchange
OpenVINO	Open Visual Inference and Neural Network Optimization
PE-PUF	Process and Environmental PUF
PH-PUF	Photonic PUF
PUFs	Physically Unclonable Functions
RF-DNA-PUF	Radio-Frequency Certificates of Authenticity
RRAM-PUF	Reconfigurable Resistive RAM PUF
RTMS	Realtime Traffic Monitoring Systems
SAC	Strict Avalanche Condition
SASE	Secure Access Service Edge
SC-PUF	ScanPUF
SDN	Software Defined Networking
SEACOD	Selective Encryption and Component-Oriented Deduplication
TERO-PUF	Transient Effect Ring Oscillator PUF
VMs	Virtual Machines
WBI	Web-Based Intermediaries

References

1. IMD. What Is the Internet of Things (IoT) & Why Is It Important? 2024. Available online: https://www.imd.org/blog/digital-transformation/internet-of-things/ (accessed on 17 September 2024).
2. Vailshery, L.S. Number of IoT Connections Worldwide 2022–2033. 2024. Available online: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/ (accessed on 17 September 2024).
3. Albreem, M.A.; Sheikh, A.M.; Alsharif, M.H.; Jusoh, M.; Mohd Yasin, M.N. Green Internet of Things (GIoT): Applications, Practices, Awareness, and Challenges. *IEEE Access* **2021**, *9*, 38833–38858. [CrossRef]
4. sptel. Future Development of IoT in Singapore—2024 & Beyond. 2024. Available online: https://sptel.com/future-development-of-iot/ (accessed on 15 September 2024).
5. Sheikh, A.M.; Islam, M.R.; Habaebi, M.H.; Kabbani, A.; Zabidi, S.A.; bin Najeeb, A.R. Securing the IoT Edge Devices Using Advanced Digital Technologies. *Asian J. Electr. Electron. Eng.* **2024**, *4*, 52–60. [CrossRef]
6. Yu, W.; Liang, F.; He, X.; Hatcher, W.G.; Lu, C.; Lin, J.; Yang, X. A survey on the edge computing for the Internet of Things. *IEEE Access* **2017**, *6*, 6900–6919. [CrossRef]
7. Alwarafy, A.; Al-Thelaya, K.A.; Abdallah, M.; Schneider, J.; Hamdi, M. A survey on security and privacy issues in edge-computing-assisted internet of things. *IEEE Internet Things J.* **2020**, *8*, 4004–4022. [CrossRef]
8. Shi, W.; Cao, J.; Zhang, Q.; Li, Y.; Xu, L. Edge computing: Vision and challenges. *IEEE Internet Things J.* **2016**, *3*, 637–646. [CrossRef]
9. Yahuza, M.; Idris, M.Y.I.B.; Wahab, A.W.B.A.; Ho, A.T.; Khan, S.; Musa, S.N.B.; Taha, A.Z.B. Systematic review on security and privacy requirements in edge computing: State of the art and future research opportunities. *IEEE Access* **2020**, *8*, 76541–76567. [CrossRef]
10. Shi, W.; Dustdar, S. The Promise of Edge Computing. *Computer* **2016**, *49*, 78–81. [CrossRef]
11. Mukherjee, M.; Matam, R.; Shu, L.; Maglaras, L.; Ferrag, M.A.; Choudhury, N.; Kumar, V. Security and privacy in fog computing: Challenges. *IEEE Access* **2017**, *5*, 19293–19304. [CrossRef]
12. Chen, B.; Wan, J.; Celesti, A.; Li, D.; Abbas, H.; Zhang, Q. Edge Computing in IoT-Based Manufacturing. *IEEE Commun. Mag.* **2018**, *56*, 103–109. [CrossRef]
13. Premsankar, G.; Di Francesco, M.; Taleb, T. Edge computing for the Internet of Things: A case study. *IEEE Internet Things J.* **2018**, *5*, 1275–1284. [CrossRef]
14. Mo, W.; Wang, T.; Zhang, S.; Zhang, J. An active and verifiable trust evaluation approach for edge computing. *J. Cloud Comput.* **2020**, *9*, 51 . [CrossRef]
15. Liao, H.; Zhou, Z.; Zhao, X.; Zhang, L.; Mumtaz, S.; Jolfaei, A.; Ahmed, S.H.; Bashir, A.K. Learning-based context-aware resource allocation for edge-computing-empowered industrial IoT. *IEEE Internet Things J.* **2019**, *7*, 4260–4277. [CrossRef]
16. Zhang, J.; Chen, B.; Zhao, Y.; Cheng, X.; Hu, F. Data security and privacy-preserving in edge computing paradigm: Survey and open issues. *IEEE Access* **2018**, *6*, 18209–18237. [CrossRef]
17. Rupanetti, D.; Kaabouch, N. Combining Edge Computing-Assisted Internet of Things Security with Artificial Intelligence: Applications, Challenges, and Opportunities. *Appl. Sci.* **2024**, *14*, 7104. [CrossRef]
18. Xiao, Y.; Jia, Y.; Liu, C.; Cheng, X.; Yu, J.; Lv, W. Edge computing security: State of the art and challenges. *Proc. IEEE* **2019**, *107*, 1608–1631. [CrossRef]
19. Begg, R. Digital Supply Chain: Cybersecurity Report Flags Clear and Present Danger. Available online: https://www.machinedesign.com/automation-iiot/article/21236851/netscout-digital-supply-chain-cybersecurity-report-flags-clear-and-present-danger/ (accessed on 17 September 2022).
20. Jovanovic, B. Internet of Things statistics for 2022—Taking Things Apart. 2022. Available online: https://dataprot.net/statistics/iot-statistics/#:~:text=Malware%20attacks%20are%20now%20affecting,a%20third%20of%20infected%20devices (accessed on 17 September 2022).
21. Yao, A.; Li, G.; Li, X.; Jiang, F.; Xu, J.; Liu, X. Differential privacy in edge computing-based smart city Applications: Security issues, solutions and future directions. *Array* **2023**, *19*, 100293. [CrossRef]
22. Rao, F.Y.; Bertino, E. Privacy techniques for edge computing systems. *Proc. IEEE* **2019**, *107*, 1632–1654. [CrossRef]
23. Lyu, M.; Ni, Z.; Chen, Q.; Li, F. Edge-DPSDG: An Edge-based Differential Privacy Protection Model for Smart Healthcare. *IEEE Trans. Big Data* **2024**, *11*, 21–34 . [CrossRef]
24. Jiang, B.; Li, J.; Wang, H.; Song, H. Privacy-preserving federated learning for industrial edge computing via hybrid differential privacy and adaptive compression. *IEEE Trans. Ind. Inform.* **2021**, *19*, 1136–1144. [CrossRef]
25. Zerraza, I.; Seghir, Z.A.; Hemam, M. An Efficient Lightweight Authentication and Access Control for IoT Edge Devices. *Int. J. Saf. Secur. Eng.* **2024**, *14*, 807–813. [CrossRef]
26. Rostampour, S.; Bagheri, N.; Bendavid, Y.; Safkhani, M.; Kumari, S.; Rodrigues, J.J. An authentication protocol for next generation of constrained Iot systems. *IEEE Internet Things J.* **2022**, *9*, 21493–21504. [CrossRef]
27. Ding, X.; Wang, X.; Xie, Y.; Li, F. A lightweight anonymous authentication protocol for resource-constrained devices in Internet of Things. *IEEE Internet Things J.* **2021**, *9*, 1818–1829. [CrossRef]

28. Wang, X.; Garg, S.; Lin, H.; Kaddoum, G.; Hu, J.; Hossain, M.S. A secure data aggregation strategy in edge computing and blockchain-empowered internet of things. *IEEE Internet Things J.* **2020**, *9*, 14237–14246. [CrossRef]
29. Wang, J.; Wu, L.; Choo, K.K.R.; He, D. Blockchain-based anonymous authentication with key management for smart grid edge computing infrastructure. *IEEE Trans. Ind. Inform.* **2019**, *16*, 1984–1992. [CrossRef]
30. Bai, F.; Shen, T.; Yu, Z.; Zeng, K.; Gong, B. Trustworthy blockchain-empowered collaborative edge computing-as-a-service scheduling and data sharing in the IIoE. *IEEE Internet Things J.* **2021**, *9*, 14752–14766. [CrossRef]
31. Li, B.; Chen, T.; Giannakis, G.B. Secure mobile edge computing in IoT via collaborative online learning. *IEEE Trans. Signal Process.* **2019**, *67*, 5922–5935. [CrossRef]
32. Wang, K.; Yin, H.; Quan, W.; Min, G. Enabling collaborative edge computing for software defined vehicular networks. *IEEE Netw.* **2018**, *32*, 112–117. [CrossRef]
33. Zhang, P.; Wang, Y.; Kumar, N.; Jiang, C.; Shi, G. A security-and privacy-preserving approach based on data disturbance for collaborative edge computing in social IoT systems. *IEEE Trans. Comput. Soc. Syst.* **2021**, *9*, 97–108. [CrossRef]
34. Bourechak, A.; Zedadra, O.; Kouahla, M.N.; Guerrieri, A.; Seridi, H.; Fortino, G. At the confluence of artificial intelligence and edge computing in iot-based applications: A review and new perspectives. *Sensors* **2023**, *23*, 1639. [CrossRef]
35. Chang, Z.; Liu, S.; Xiong, X.; Cai, Z.; Tu, G. A survey of recent advances in edge-computing-powered artificial intelligence of things. *IEEE Internet Things J.* **2021**, *8*, 13849–13875. [CrossRef]
36. Deng, S.; Zhao, H.; Fang, W.; Yin, J.; Dustdar, S.; Zomaya, A.Y. Edge intelligence: The confluence of edge computing and artificial intelligence. *IEEE Internet Things J.* **2020**, *7*, 7457–7469. [CrossRef]
37. Manan, A. Implementation of image processing algorithm on FPGA. *Akgec J. Technol.* **2006**, *2*, 25–28.
38. Biookaghazadeh, S.; Zhao, M.; Ren, F. Are {FPGAs} Suitable for Edge Computing? In Proceedings of the USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18), Boston, MA, USA, 10 July 2018.
39. Sipola, T.; Alatalo, J.; Kokkonen, T.; Rantonen, M. Artificial intelligence in the IoT era: A review of edge AI hardware and software. In Proceedings of the 2022 31st Conference of Open Innovations Association (FRUCT), Helsinki, Finland, 27–29 April 2022; pp. 320–331.
40. Jiang, S.; He, D.; Yang, C.; Xu, C.; Luo, G.; Chen, Y.; Liu, Y.; Jiang, J. Accelerating mobile applications at the network edge with software-programmable FPGAs. In Proceedings of the IEEE INFOCOM 2018-IEEE Conference on Computer Communications, Honolulu, HI, USA, 16–19 April 2018; pp. 55–62.
41. Zhao, H.; Ratazzi, P. A Lightweight Hardware-Assisted Security Method for eFPGA Edge Devices. *IEEE Internet Things J.* **2024**, *11*, 23673–23682. [CrossRef]
42. Ngo, D.M.; Temko, A.; Murphy, C.C.; Popovici, E. FPGA hardware acceleration framework for anomaly-based intrusion detection system in IoT. In Proceedings of the 2021 31st International Conference on Field-Programmable Logic and Applications (FPL), Dresden, Germany, 30 August–3 September 2021; pp. 69–75.
43. Zhang, Y.; Zhu, M.; Yang, B.; Liu, L. A Highly Reliable Strong Physical Unclonable Function Design Based on FPGA. *J. Phys. Conf. Ser.* **2020**, *1619*, 012003. [CrossRef]
44. Xu, T.; Potkonjak, M. Robust and flexible FPGA-based digital PUF. In Proceedings of the 2014 24th International Conference on Field Programmable Logic and Applications (FPL), Munich, Germany, 2–4 September 2014; pp. 1–6.
45. Guajardo, J.; Kumar, S.S.; Schrijen, G.J.; Tuyls, P. FPGA intrinsic PUFs and their use for IP protection. In Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems, Vienna, Austria, 10–13 September 2007; pp. 63–80.
46. Kang, H.; Hori, Y.; Satoh, A. Performance evaluation of the first commercial PUF-embedded RFID. In Proceedings of the The 1st IEEE Global Conference on Consumer Electronics, Tokyo, Japan, 2–5 October 2012; pp. 5–8.
47. Suh, G.E.; Devadas, S. Physical unclonable functions for device authentication and secret key generation. In Proceedings of the 2007 44th ACM/IEEE Design Automation Conference, San Diego, CA, USA, 4–8 June 2007; pp. 9–14.
48. Alkabani, Y.; Koushanfar, F.; Potkonjak, M. Remote activation of ICs for piracy prevention and digital right management. In Proceedings of the 2007 IEEE/ACM International Conference on Computer-Aided Design, San Jose, CA, USA, 4–8 November 2007; pp. 674–677.
49. Liu, D.; Yan, Z.; Ding, W.; Atiquzzaman, M. A survey on secure data analytics in edge computing. *IEEE Internet Things J.* **2019**, *6*, 4946–4967. [CrossRef]
50. Xu, Z.; Liu, W.; Huang, J.; Yang, C.; Lu, J.; Tan, H. Artificial intelligence for securing IoT services in edge computing: A survey. *Secur. Commun. Netw.* **2020**, *2020*, 8872586. [CrossRef]
51. Lee, J.H.; Kim, H. Security and privacy challenges in the internet of things [security and privacy matters]. *IEEE Consum. Electron. Mag.* **2017**, *6*, 134–136. [CrossRef]
52. Anandakumar, N.N.; Hashmi, M.S.; Tehranipoor, M. FPGA-based Physical Unclonable Functions: A comprehensive overview of theory and architectures. *Integration* **2021**, *81*, 175–194. [CrossRef]
53. Lounis, K.; Zulkernine, M. More Lessons: Analysis of PUF-Based Authentication Protocols for IoT. Cryptology ePrint Archive, Paper 2021/1509. 2021. Available online: https://eprint.iacr.org/2021/1509 (accessed on 14 February 2025).

54. Majzoobi, M.; Koushanfar, F.; Potkonjak, M.; Tehranipoor, M.; Wang, C. FPGA-oriented Security. In *Introduction to Hardware Security and Trust*; Springer: New York, NY, USA, 2011; pp. 195–231.
55. Merenda, M.; Porcaro, C.; Iero, D. Edge machine learning for ai-enabled iot devices: A review. *Sensors* **2020**, *20*, 2533. [CrossRef]
56. Ansari, M.S.; Alsamhi, S.H.; Qiao, Y.; Ye, Y.; Lee, B. Security of distributed intelligence in edge computing: Threats and countermeasures. In *The Cloud-to-Thing Continuum*; Palgrave Macmillan: Cham, Switzerland, 2020; pp. 95–122.
57. El-Saleh, A.A.; Sheikh, A.M.; Albreem, M.A.; Honnurvali, M.S. The Internet of Medical Things (IoMT): Opportunities and challenges. *Wirel. Netw.* **2024**, *31*, 327–344. [CrossRef]
58. Liu, H.; Zhang, Y.; Yang, T. Blockchain-Enabled Security in Electric Vehicles Cloud and Edge Computing. *IEEE Netw.* **2018**, *32*, 78–83. [CrossRef]
59. Bonnah, E.; Shiguang, J. DecChain: A decentralized security approach in Edge Computing based on Blockchain. *Future Gener. Comput. Syst.* **2020**, *113*, 363–379. [CrossRef]
60. Yang, R.; Yu, F.R.; Si, P.; Yang, Z.; Zhang, Y. Integrated Blockchain and Edge Computing Systems: A Survey, Some Research Issues and Challenges. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 1508–1532. [CrossRef]
61. Luo, C.; Xu, L.; Li, D.; Wu, W. Edge computing integrated with blockchain technologies. In *Complexity and Approximation*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 268–288.
62. Mendki, P. Blockchain enabled iot edge computing: Addressing privacy, security and other challenges. In Proceedings of the 2020 The 2nd International Conference on Blockchain Technology, Hilo, HI, USA, 12–14 March 2020; pp. 63–67.
63. Kotevska, O.; Johnson, J.; Kusne, A.G. Analyzing Data Privacy for Edge Systems. In Proceedings of the 2022 IEEE International Conference on Smart Computing (SMARTCOMP), Espoo, Finland, 20–24 June 2022; pp. 223–228.
64. Paillet, D. *An Overview of Cybersecurity Best Practices for Edge Computing*; Technical Report; Schneider Electric: Rueil-Malmaison, France, 2021.
65. Johannes Beekman. How to Deal with Edge Computing Security Concerns. 2022. Available online: https://iotmktg.com/how-to-deal-edge-computing-security-concerns/ (accessed on 17 September 2024).
66. van der Walt, S.; Venter, H. Research gaps and opportunities for secure access service edge. In Proceedings of the International Conference on Cyber Warfare and Security, Albany, NY, USA, 17–18 March 2022; Volume 17, pp. 609–619.
67. Singh, S.; Sulthana, R.; Shewale, T.; Chamola, V.; Benslimane, A.; Sikdar, B. Machine-Learning-Assisted Security and Privacy Provisioning for Edge Computing: A Survey. *IEEE Internet Things J.* **2022**, *9*, 236–260. [CrossRef]
68. Waguie, F.T.; Al-Turjman, F. Artificial intelligence for edge computing security: A survey. In Proceedings of the 2022 International Conference on Artificial Intelligence in Everything (AIE), Lefkosa, Cyprus, 2–4 August 2022; pp. 446–450.
69. Zhou, Z.; Chen, X.; Li, E.; Zeng, L.; Luo, K.; Zhang, J. Edge intelligence: Paving the last mile of artificial intelligence with edge computing. *Proc. IEEE* **2019**, *107*, 1738–1762. [CrossRef]
70. Lin, F.; Zhou, Y.; An, X.; You, I.; Choo, K.K.R. Fair resource allocation in an intrusion-detection system for edge computing: Ensuring the security of Internet of Things devices. *IEEE Consum. Electron. Mag.* **2018**, *7*, 45–50. [CrossRef]
71. Spadaccino, P.; Cuomo, F. Intrusion Detection Systems for IoT: Opportunities and challenges offered by Edge Computing and Machine Learning. *arXiv* **2020**, arXiv:2012.01174.
72. Singh, A.; Chatterjee, K.; Satapathy, S.C. An edge based hybrid intrusion detection framework for mobile edge computing. *Complex Intell. Syst.* **2022**, *8*, 3719–3746. [CrossRef]
73. Alsubhi, K. A Secured Intrusion Detection System for Mobile Edge Computing. *Appl. Sci.* **2024**, *14*, 1432. [CrossRef]
74. Gyamfi, E.; Jurcut, A. Intrusion detection in internet of things systems: A review on design approaches leveraging multi-access edge computing, machine learning, and datasets. *Sensors* **2022**, *22*, 3744. [CrossRef]
75. Long, J.; Liang, W.; Li, K.C.; Zhang, D.; Tang, M.; Luo, H. PUF-based anonymous authentication scheme for hardware devices and IPs in edge computing environment. *IEEE Access* **2019**, *7*, 124785–124796. [CrossRef]
76. Aarella, S.G.; Mohanty, S.P.; Kougianos, E.; Puthal, D. PUF-based Authentication Scheme for Edge Data Centers in Collaborative Edge Computing. In Proceedings of the 2022 IEEE International Symposium on Smart Electronic Systems (iSES), Warangal, India, 18–20 December 2022; pp. 433–438.
77. Chen, Z.; Wu, M.; Zhou, Y.; Li, R.; Tan, J.; Ding, D. Puf-cim: Sram-based compute-in-memory with zero bit-error-rate physical unclonable function for lightweight secure edge computing. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2023**, *31*, 1234–1247. [CrossRef]
78. Zhang, J.L.; Wu, Q.; Ding, Y.P.; Lv, Y.Q.; Zhou, Q.; Xia, Z.H.; Sun, X.M.; Wang, X.W. Techniques for design and implementation of an FPGA-specific physical unclonable function. *J. Comput. Sci. Technol.* **2016**, *31*, 124–136. [CrossRef]
79. Zhu, C.; Zhu, X.; Ren, J.; Qin, T. Blockchain-enabled federated learning for UAV edge computing network: Issues and solutions. *IEEE Access* **2022**, *10*, 56591–56610. [CrossRef]
80. Nguyen, T.; Nguyen, H.; Gia, T.N. Exploring the integration of edge computing and blockchain IoT: Principles, architectures, security, and applications. *J. Netw. Comput. Appl.* **2024**, *226*, 103884. [CrossRef]

81. Song, S.; Choi, B.Y.; Kim, D. Selective encryption and component-oriented deduplication for mobile cloud data computing. In Proceedings of the 2016 International Conference on Computing, Networking and Communications (ICNC), Kauai, HI, USA, 15–18 February 2016; pp. 1–5.
82. Zhao, Y.; Wang, W.; Li, Y.; Meixner, C.C.; Tornatore, M.; Zhang, J. Edge computing and networking: A survey on infrastructures and applications. *IEEE Access* **2019**, *7*, 101213–101230. [CrossRef]
83. Cheng, G.; Chen, Y.; Deng, S.; Gao, H.; Yin, J. A Blockchain-Based Mutual Authentication Scheme for Collaborative Edge Computing. *IEEE Trans. Comput. Soc. Syst.* **2022**, *9*, 146–158. [CrossRef]
84. Zeyu, H.; Geming, X.; Zhaohang, W.; Sen, Y. Survey on Edge Computing Security. In Proceedings of the 2020 International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE), Fuzhou, China, 12–14 June 2020; pp. 96–105. [CrossRef]
85. Mollah, M.B.; Azad, M.A.K.; Vasilakos, A. Secure data sharing and searching at the edge of cloud-assisted internet of things. *IEEE Cloud Comput.* **2017**, *4*, 34–42. [CrossRef]
86. Bhat, S.A.; Sofi, I.B.; Chi, C.Y. Edge computing and its convergence with blockchain in 5G and beyond: Security, challenges, and opportunities. *IEEE Access* **2020**, *8*, 205340–205373. [CrossRef]
87. Microsoft. What Is Edge Computing? 2022. Available online: https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-edge-computing/ (accessed on 18 October 2022).
88. Mohanan, R. What Is Edge Computing? Components, Examples, and Best Practices. 2022. Available online: https://www.spiceworks.com/tech/edge-computing/articles/what-is-edge-computing/ (accessed on 18 October 2022).
89. Hemminger, S. Network emulation with NetEm. In Proceedings of the Linux conf au, Canberra, ACT, Australia, 18–23 April 2005; Volume 5, p. 2005.
90. Li, W.; Chen, Z.; Gao, X.; Liu, W.; Wang, J. Multimodel framework for indoor localization under mobile edge computing environment. *IEEE Internet Things J.* **2018**, *6*, 4844–4853. [CrossRef]
91. Khan, W.Z.; Ahmed, E.; Hakak, S.; Yaqoob, I.; Ahmed, A. Edge computing: A survey. *Future Gener. Comput. Syst.* **2019**, *97*, 219–235. [CrossRef]
92. Toczé, K.; Nadjm-Tehrani, S. A taxonomy for management and optimization of multiple resources in edge computing. *Wirel. Commun. Mob. Comput.* **2018**, *2018*, 7476201. [CrossRef]
93. Albreem, M.A.; Sheikh, A.M.; Bashir, M.J.; El-Saleh, A.A. Towards green Internet of Things (IoT) for a sustainable future in Gulf Cooperation Council countries: Current practices, challenges and future prospective. *Wirel. Netw.* **2023**, *29*, 539–567. [CrossRef]
94. Varghese, B.; Wang, N.; Barbhuiya, S.; Kilpatrick, P.; Nikolopoulos, D.S. Challenges and opportunities in edge computing. In Proceedings of the 2016 IEEE International Conference on Smart Cloud (SmartCloud), New York, NY, USA, 18–20 November 2016; pp. 20–26.
95. Sunku, R. Challenges in Edge Computing. 2022. Available online: https://sunkur.medium.com/challenges-in-edge-computing-ec9237b5ab77 (accessed on 12 November 2022).
96. Kong, L.; Tan, J.; Huang, J.; Chen, G.; Wang, S.; Jin, X.; Zeng, P.; Khan, M.K.; Das, S.K. Edge-Computing-Driven Internet of Things: A Survey. *ACM Comput. Surv. (CSUR)* **2022**, *55*, 174. [CrossRef]
97. Schäfer, D.; Edinger, J.; VanSyckel, S.; Paluska, J.M.; Becker, C. Tasklets: Overcoming heterogeneity in distributed computing systems. In Proceedings of the 2016 IEEE 36th International Conference on Distributed Computing Systems Workshops (ICDCSW), Nara, Japan, 27–30 June 2016; pp. 156–161.
98. Carvalho, G.; Cabral, B.; Pereira, V.; Bernardino, J. Edge computing: Current trends, research challenges and future directions. *Computing* **2021**, *103*, 993–1023. [CrossRef]
99. Sun, J.; Gu, Q.; Zheng, T.; Dong, P.; Qin, Y. Joint communication and computing resource allocation in vehicular edge computing. *Int. J. Distrib. Sens. Netw.* **2019**, *15*, 1550147719837859. [CrossRef]
100. Jia, M.; Zhang, L.; Wu, J.; Guo, Q.; Gu, X. Joint computing and communication resource allocation for edge computing towards Huge LEO networks. *China Commun.* **2022**, *19*, 73–84. [CrossRef]
101. Ning, Z.; Huang, J.; Wang, X.; Rodrigues, J.J.; Guo, L. Mobile edge computing-enabled internet of vehicles: Toward energy-efficient scheduling. *IEEE Netw.* **2019**, *33*, 198–205. [CrossRef]
102. Wang, X.; Han, Y.; Leung, V.C.; Niyato, D.; Yan, X.; Chen, X. Convergence of edge computing and deep learning: A comprehensive survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 869–904. [CrossRef]
103. Chen, H.; Qin, W.; Wang, L. Task partitioning and offloading in IoT cloud-edge collaborative computing framework: A survey. *J. Cloud Comput.* **2022**, *11*, 1–19. [CrossRef]
104. Saeik, F.; Avgeris, M.; Spatharakis, D.; Santi, N.; Dechouniotis, D.; Violos, J.; Leivadeas, A.; Athanasopoulos, N.; Mitton, N.; Papavassiliou, S. Task offloading in Edge and Cloud Computing: A survey on mathematical, artificial intelligence and control theory solutions. *Comput. Netw.* **2021**, *195*, 108177. [CrossRef]

105. Feng, M.; Krunz, M.; Zhang, W. Task partitioning and user association for latency minimization in mobile edge computing networks. In Proceedings of the IEEE INFOCOM 2021-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS),Vancouver, BC, Canada, 10–13 May 2021; pp. 1–6.
106. Ometov, A.; Molua, O.L.; Komarov, M.; Nurmi, J. A survey of security in cloud, edge, and fog computing. *Sensors* **2022**, *22*, 927. [CrossRef]
107. Sha, K.; Yang, T.A.; Wei, W.; Davari, S. A survey of edge computing-based designs for IoT security. *Digit. Commun. Netw.* **2020**, *6*, 195–202. [CrossRef]
108. Ahmadi, S. Security Implications of Edge Computing in Cloud Networks. *J. Comput. Commun.* **2024**, *12*, 26–46. [CrossRef]
109. Accenture. Edge Computing. 2024. Available online: https://www.accenture.com/bg-en/insights/cloud/edge-computing-index (accessed on 15 September 2024).
110. Abhishek, A.; Adeniyi-Jones, C.; Van Hensbergen, E.; Balmakhtar, M. Accounting and resource scheduling at the edge. In Proceedings of the HotEdge'20, Online, 25–26 June 2020. Available online: https://www.usenix.org/system/files/hotedge20_poster_abhishek.pdf (accessed on 27 September 2024).
111. Ahmed, E.; Ahmed, A.; Yaqoob, I.; Shuja, J.; Gani, A.; Imran, M.; Shoaib, M. Bringing Computation Closer toward the User Network: Is Edge Computing the Solution? *IEEE Commun. Mag.* **2017**, *55*, 138–144. [CrossRef]
112. Umme Sutarwala, Edge Computing: Four Cybersecurity Challenges in 2022. 7 March 2022. Available online: https://itsecuritywire.com/featured/edge-computing-four-cybersecurity-challenges-in-2022/ (accessed on 8 February 2023).
113. Wu, W.; Zhang, Q.; Wang, H.J. Edge computing security protection from the perspective of classified protection of cybersecurity. In Proceedings of the 2019 6th International Conference on Information Science and Control Engineering (ICISCE), Shanghai, China, 20–22 December 2019; pp. 278–281.
114. Mall, A.; Singh, P.; Thute, A.; Khapre, S.P.; Shankar, A. Security issues of edge computing in IoT. In Proceedings of the International Conference on Machine Intelligence and Data Science Applications: MIDAS 2020, Dehradun, India, 4–5 September 2021; pp. 567–579.
115. Ranaweera, P.; Jurcut, A.D.; Liyanage, M. Survey on multi-access edge computing security and privacy. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 1078–1124. [CrossRef]
116. Uddin, R.; Kumar, S.A.; Chamola, V. Denial of service attacks in edge computing layers: Taxonomy, vulnerabilities, threats and solutions. *Ad Hoc Netw.* **2024**, *152*, 103322. [CrossRef]
117. Fazeldehkordi, E.; Grønli, T.M. A Survey of Security Architectures for Edge Computing-Based IoT. *IoT* **2022**, *3*, 332–365. [CrossRef]
118. Hassan, N.; Gillani, S.; Ahmed, E.; Yaqoob, I.; Imran, M. The Role of Edge Computing in Internet of Things. *IEEE Commun. Mag.* **2018**, *56*, 110–115. [CrossRef]
119. Alwakeel, A.M. An overview of fog computing and edge computing security and privacy issues. *Sensors* **2021**, *21*, 8226. [CrossRef]
120. Guynes, S.; Parrish, J.; Vedder, R. Edge computing societal privacy and security issues. *ACM SIGCAS Comput. Soc.* **2020**, *48*, 11–12. [CrossRef]
121. Mukherjee, M.; Matam, R.; Mavromoustakis, C.X.; Jiang, H.; Mastorakis, G.; Guo, M. Intelligent edge computing: Security and privacy challenges. *IEEE Commun. Mag.* **2020**, *58*, 26–31. [CrossRef]
122. Zhu, W.; Zhou, C.; Jiang, L. A Trusted Internet of Things Access Scheme for Cloud Edge Collaboration. *Electronics* **2024**, *13*, 1026. [CrossRef]
123. Brett Daniel Is Edge Computing Secure? 9 December 2020. Available online: https://www.trentonsystems.com/blog/is-edge-computing-secure (accessed on 14 January 2023).
124. Xia, Q.; Tao, Z.; Li, Q. Privacy issues in edge computing. In *Fog/Edge Computing For Security, Privacy, and Applications*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 147–169.
125. Adil, M.; Almaiah, M.A.; Omar Alsayed, A.; Almomani, O. An anonymous channel categorization scheme of edge nodes to detect jamming attacks in wireless sensor networks. *Sensors* **2020**, *20*, 2311. [CrossRef] [PubMed]
126. Liang, F.; Hatcher, W.G.; Liao, W.; Gao, W.; Yu, W. Machine learning for security and the internet of things: The good, the bad, and the ugly. *IEEE Access* **2019**, *7*, 158126–158147. [CrossRef]
127. Sheikh, A.M.; Islam, M.R.; Habaebi, M.H.; Zabidi, S.A.; Najeeb, A.R.B.; Basahel, A. Machine Learning (ML) assisted Edge security framework on FPGAs. In Proceedings of the 2023 9th International Conference on Computer and Communication Engineering (ICCCE), Kualalumpur, Malaysia, 15–16 August 2023; pp. 155–160.
128. Blech, R. Data Encryption for the Edge. 6 October 2021. Available online: https://www.xsoccorp.com/post/data-encryption-for-the-edge (accessed on 19 June 2023).
129. Kwon, H.; Hahn, C.; Kim, D.; Hur, J. Secure deduplication for multimedia data with user revocation in cloud storage. *Multimed. Tools Appl.* **2017**, *76*, 5889–5903. [CrossRef]
130. Neto, E.C.P.; Dadkhah, S.; Ghorbani, A.A. Sustainable and Secure Optimization of Load Distribution in Edge Computing. In Proceedings of the 2022 IEEE 19th International Conference on Smart Communities: Improving Quality of Life Using ICT, IoT and AI (HONET), Marietta, GA, USA, 10–12 October 2022; pp. 040–045.

131. Li, D.; Zhang, E.; Lei, M.; Song, C. Zero trust in edge computing environment: A blockchain based practical scheme. *Math. Biosci. Eng.* **2022**, *19*, 4196–4216. [CrossRef]
132. Wu, Y.; Dai, H.N.; Wang, H. Convergence of blockchain and edge computing for secure and scalable IIoT critical infrastructures in industry 4.0. *IEEE Internet Things J.* **2020**, *8*, 2300–2317. [CrossRef]
133. Dong, J.; Song, C.; Zhang, T.; Li, Y.; Zheng, H. Integration of edge computing and blockchain for provision of data fusion and secure big data analysis for Internet of Things. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 9233267. [CrossRef]
134. Ding, X.; Guo, J.; Li, D.; Wu, W. Pricing and budget allocation for IoT blockchain with edge computing. *IEEE Trans. Cloud Comput.* **2022**, *11*, 1608–1621. [CrossRef]
135. Medhane, D.V.; Sangaiah, A.K.; Hossain, M.S.; Muhammad, G.; Wang, J. Blockchain-enabled distributed security framework for next-generation IoT: An edge cloud and software-defined network-integrated approach. *IEEE Internet Things J.* **2020**, *7*, 6143–6149. [CrossRef]
136. Wang, S.; Liu, Z.; Wang, H.; Wang, J. Ensuring security in edge computing through effective blockchain node detection. *J. Cloud Comput.* **2023**, *12*, 88. [CrossRef]
137. Xu, X.; Zhang, X.; Gao, H.; Xue, Y.; Qi, L.; Dou, W. BeCome: Blockchain-enabled computation offloading for IoT in mobile edge computing. *IEEE Trans. Ind. Inform.* **2019**, *16*, 4187–4195. [CrossRef]
138. Jangirala, S.; Das, A.K.; Vasilakos, A.V. Designing secure lightweight blockchain-enabled RFID-based authentication protocol for supply chains in 5G mobile edge computing environment. *IEEE Trans. Ind. Inform.* **2019**, *16*, 7081–7093. [CrossRef]
139. Gao, Q.; Xiao, J.; Cao, Y.; Deng, S.; Ouyang, C.; Feng, Z. Blockchain-based collaborative edge computing: Efficiency, incentive and trust. *J. Cloud Comput.* **2023**, *12*, 72. [CrossRef]
140. Mandarino, V.; Pappalardo, G.; Tramontana, E. A Blockchain-Based Electronic Health Record (EHR) System for Edge Computing Enhancing Security and Cost Efficiency. *Computers* **2024**, *13*, 132. [CrossRef]
141. Xiong, Z.; Zhang, Y.; Niyato, D.; Wang, P.; Han, Z. When mobile blockchain meets edge computing. *IEEE Commun. Mag.* **2018**, *56*, 33–39. [CrossRef]
142. Mahadevappa, P.; Murugesan, R.K.; Al-Amri, R.; Thabit, R.; Al-Ghushami, A.H.; Alkawsi, G. A secure edge computing model using machine learning and IDS to detect and isolate intruders. *MethodsX* **2024**, *12*, 102597. [CrossRef]
143. Meuser, T.; Lovén, L.; Bhuyan, M.; Patil, S.G.; Dustdar, S.; Aral, A.; Bayhan, S.; Becker, C.; de Lara, E.; Ding, A.Y.; et al. Revisiting Edge AI: Opportunities and Challenges. *IEEE Internet Comput.* **2024**, *28*, 49–59. [CrossRef]
144. Mendez, J.; Bierzynski, K.; Cuéllar, M.; Morales, D.P. Edge Intelligence: Concepts, Architectures, Applications, and Future Directions. *ACM Trans. Embed. Comput. Syst. (TECS)* **2022**, *21*, 1–41. [CrossRef]
145. Wang, C.; Yuan, Z.; Zhou, P.; Xu, Z.; Li, R.; Wu, D.O. The security and privacy of mobile edge computing: An artificial intelligence perspective. *IEEE Internet Things J.* **2023**, *10*, 22008–22032. [CrossRef]
146. Xu, D.; Li, T.; Li, Y.; Su, X.; Tarkoma, S.; Jiang, T.; Crowcroft, J.; Hui, P. Edge intelligence: Empowering intelligence to the edge of network. *Proc. IEEE* **2021**, *109*, 1778–1837. [CrossRef]
147. Raimundo, R.; Rosário, A. The impact of artificial intelligence on data system security: A literature review. *Sensors* **2021**, *21*, 7029. [CrossRef] [PubMed]
148. Montini, H. Artificial Intelligence in Cybersecurity: How to Use the Technology. 2024. Available online: https://www.provendata.com/blog/ai-in-cybersecurity/ (accessed on 8 October 2024).
149. Alowais, S.A.; Alghamdi, S.S.; Alsuhebany, N.; Alqahtani, T.; Alshaya, A.I.; Almohareb, S.N.; Aldairem, A.; Alrashed, M.; Bin Saleh, K.; Badreldin, H.A.; et al. Revolutionizing healthcare: The role of artificial intelligence in clinical practice. *BMC Med. Educ.* **2023**, *23*, 689. [CrossRef] [PubMed]
150. Wang, F.; Zhang, M.; Wang, X.; Ma, X.; Liu, J. Deep learning for edge computing applications: A state-of-the-art survey. *IEEE Access* **2020**, *8*, 58322–58336. [CrossRef]
151. Sun, Y.; Ochiai, H.; Esaki, H. Decentralized deep learning for multi-access edge computing: A survey on communication efficiency and trustworthiness. *IEEE Trans. Artif. Intell.* **2021**, *3*, 963–972. [CrossRef]
152. Wu, H.; Han, H.; Wang, X.; Sun, S. Research on artificial intelligence enhancing internet of things security: A survey. *IEEE Access* **2020**, *8*, 153826–153848. [CrossRef]
153. Kulin, M.; Kazaz, T.; De Poorter, E.; Moerman, I. A Survey on Machine Learning-Based Performance Improvement of Wireless Networks: PHY, MAC and Network Layer. *Electronics* **2021**, *10*, 318. [CrossRef]
154. Hua, H.; Li, Y.; Wang, T.; Dong, N.; Li, W.; Cao, J. Edge Computing with Artificial Intelligence: A Machine Learning Perspective. *ACM Comput. Surv.* **2023**, *55*, 1–35. [CrossRef]
155. Munir, A.; Blasch, E.; Kwon, J.; Kong, J.; Aved, A. Artificial intelligence and data fusion at the edge. *IEEE Aerosp. Electron. Syst. Mag.* **2021**, *36*, 62–78. [CrossRef]
156. Sedjelmaci, H.; Senouci, S.M.; Ansari, N.; Boualouache, A. A trusted hybrid learning approach to secure edge computing. *IEEE Consum. Electron. Mag.* **2021**, *11*, 30–37. [CrossRef]

157. Gassend, B.; Clarke, D.; Van Dijk, M.; Devadas, S. Silicon physical random functions. In Proceedings of the 9th ACM Conference on Computer and Communications Security, Washington, DC USA, 18–22 November 2002; pp. 148–160.
158. Shamsoshoara, A.; Korenda, A.; Afghah, F.; Zeadally, S. A survey on physical unclonable function (PUF)-based security solutions for Internet of Things. *Comput. Netw.* **2020**, *183*, 107593. [CrossRef]
159. Murphy, D. The Role of Machine Learning in Data Security. 2023. Available online: https://www.lepide.com/blog/the-role-of-machine-learning-in-data-security/ (accessed on 24 January 2024).
160. Rigaki, M.; Garcia, S. A survey of privacy attacks in machine learning. *ACM Comput. Surv.* **2023**, *56*, 1–34. [CrossRef]
161. Shen, T.; Ding, L.; Sun, J.; Jing, C.; Guo, F.; Wu, C. Edge Computing for IoT Security: Integrating Machine Learning with Key Agreement. In Proceedings of the 2023 3rd International Conference on Consumer Electronics and Computer Engineering (ICCECE), Guangzhou, China, 6–8 January 2023; pp. 474–483. [CrossRef]
162. Nassif, A.B.; Talib, M.A.; Nasir, Q.; Dakalbab, F.M. Machine learning for anomaly detection: A systematic review. *IEEE Access* **2021**, *9*, 78658–78700. [CrossRef]
163. Murshed, M.S.; Murphy, C.; Hou, D.; Khan, N.; Ananthanarayanan, G.; Hussain, F. Machine learning at the network edge: A survey. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–37. [CrossRef]
164. Brecko, A.; Kajati, E.; Koziorek, J.; Zolotova, I. Federated learning for edge computing: A survey. *Appl. Sci.* **2022**, *12*, 9124. [CrossRef]
165. Abreha, H.G.; Hayajneh, M.; Serhani, M.A. Federated learning in edge computing: A systematic survey. *Sensors* **2022**, *22*, 450. [CrossRef] [PubMed]
166. Feng, Y.; Qi, Y.; Li, H.; Wang, X.; Tian, J. Leveraging federated learning and edge computing for recommendation systems within cloud computing networks. In Proceedings of the Third International Symposium on Computer Applications and Information Systems (ISCAIS 2024), Wuhan, China, 22–24 March 2024; Volume 13210, pp. 279–287.
167. Li, X.; Wu, W. Recent Advances of Blockchain and Its Applications. *J. Soc. Comput.* **2022**, *3*, 363–394. [CrossRef]
168. Moore, E.; Imteaj, A.; Rezapour, S.; Amini, M.H. A Survey on Secure and Private Federated Learning Using Blockchain: Theory and Application in Resource-constrained Computing. *IEEE Internet Things J.* **2023**, *10*, 21942–21958. [CrossRef]
169. Ni, S.; He, Y.; Chen, L.; Wang, Y.; Yu, F. A Survey of Edge Computing Resource Allocation Strategies Based on Federated Learning. In Proceedings of the 2023 International Conference on Networking and Network Applications (NaNA), Qingdao, China, 18–21 August 2023; pp. 116–121. [CrossRef]
170. Wang, R.; Lai, J.; Zhang, Z.; Li, X.; Vijayakumar, P.; Karuppiah, M. Privacy-preserving federated learning for internet of medical things under edge computing. *IEEE J. Biomed. Health Inform.* **2022**, *27*, 854–865. [CrossRef]
171. Mahbub, M.; Gazi, M.S.A.; Provat, S.A.A.; Islam, M.S. Multi-access edge computing-aware internet of things: MEC-IoT. In Proceedings of the 2020 Emerging Technology in Computing, Communication and Electronics (ETCCE), Dhaka, Bangladesh, 21–22 December 2020; pp. 1–6.
172. Zhang, P.; Durresi, M.; Durresi, A. Mobile privacy protection enhanced with multi-access edge computing. In Proceedings of the 2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA), Krakow, Poland, 16–18 May 2018; pp. 724–731.
173. Mao, Y.; You, C.; Zhang, J.; Huang, K.; Letaief, K.B. A survey on mobile edge computing: The communication perspective. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 2322–2358. [CrossRef]
174. Zhang, Y.; Zhang, Y. The Future of Mobile Edge Computing. In *Mobile Edge Computing*; Springer: Cham, Switzerland, 2022; pp. 81–105.
175. Ortega-Fernandez, I.; Martinez, S.E.K.; Orellana, L.A.; Soldatos, J.; Kyriazis, D. Large Scale Data Anonymisation for GDPR Compliance. In *Big Data and Artificial Intelligence in Digital Finance*; Springer: Cham, Switzerland, 2022; p. 325.
176. GDPR *General Data Protection Regulation (GDPR)*; Intersoft Consulting: Hamburg, Germany, 2018; *Art. 24* . Available online: https://gdpr-info.eu/art-24-gdpr/ (accessed on 15 October 2024)
177. Sensitive Data Protection. Generalization and Bucketing. Available online: https://cloud.google.com/dlp/docs/concepts-bucketing#:~:text=Generalization%20is%20the%20process%20of,depending%20on%20the%20data%20type (accessed on 1 February 2024).
178. Khraisat, A.; Gondal, I.; Vamplew, P.; Kamruzzaman, J. Survey of intrusion detection systems: Techniques, datasets and challenges. *Cybersecurity* **2019**, *2*, 1–22. [CrossRef]
179. Cybersecurity, A. Intrusion Detection Systems (IDS) Explained. 2024. Available online: https://cybersecurity.att.com/solutions/intrusion-detection-system/ids-explained (accessed on 17 September 2024).
180. Rbah, Y.; Mahfoudi, M.; Balboul, Y.; Fattah, M.; Mazer, S.; Elbekkali, M.; Bernoussi, B. Machine learning and deep learning methods for intrusion detection systems in iomt: A survey. In Proceedings of the 2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), Meknes, Morocco, 3–4 March 2022; pp. 1–9.
181. Kikissagbe, B.R.; Adda, M. Machine learning-based intrusion detection methods in IoT systems: A comprehensive review. *Electronics* **2024**, *13*, 3601. [CrossRef]

182. Azimjonov, J.; Kim, T. Designing accurate lightweight intrusion detection systems for IoT networks using fine-tuned linear SVM and feature selectors. *Comput. Secur.* **2024**, *137*, 103598. [CrossRef]
183. Belenguer, A.; Navaridas, J.; Pascual, J.A. A review of federated learning in intrusion detection systems for IoT. *arXiv* **2022**, arXiv:2204.12443.
184. Sicato, J.C.S.; Singh, S.K.; Rathore, S.; Park, J.H. A comprehensive analyses of intrusion detection system for IoT environment. *J. Inf. Process. Syst.* **2020**, *16*, 975–990.
185. Arisdakessian, S.; Wahab, O.A.; Mourad, A.; Otrok, H.; Guizani, M. A survey on IoT intrusion detection: Federated learning, game theory, social psychology, and explainable AI as future directions. *IEEE Internet Things J.* **2022**, *10*, 4059–4092. [CrossRef]
186. Sommestad, T.; Holm, H.; Steinvall, D. Variables influencing the effectiveness of signature-based network intrusion detection systems. *Inf. Secur. J. Glob. Perspect.* **2022**, *31*, 711–728. [CrossRef]
187. Díaz-Verdejo, J.; Muñoz-Calle, J.; Estepa Alonso, A.; Estepa Alonso, R.; Madinabeitia, G. On the detection capabilities of signature-based intrusion detection systems in the context of web attacks. *Appl. Sci.* **2022**, *12*, 852. [CrossRef]
188. Nawaal, B.; Haider, U.; Khan, I.U.; Fayaz, M. Signature-based intrusion detection system for IoT. In *Cyber Security for Next-Generation Computing Technologies*; CRC Press: Boca Raton, FL, USA, 2024; pp. 141–158.
189. Mohy-Eddine, M.; Guezzaz, A.; Benkirane, S.; Azrour, M.; Farhaoui, Y. An ensemble learning based intrusion detection model for industrial IoT security. *Big Data Min. Anal.* **2023**, *6*, 273–287. [CrossRef]
190. Seng, S.; Garcia-Alfaro, J.; Laarouchi, Y. Why anomaly-based intrusion detection systems have not yet conquered the industrial market? In Proceedings of the International Symposium on Foundations and Practice of Security, Ottawa, ON, Canada, 12–14 December 2021; pp. 341–354.
191. Tufan, E.; Tezcan, C.; Acartürk, C. Anomaly-based intrusion detection by machine learning: A case study on probing attacks to an institutional network. *IEEE Access* **2021**, *9*, 50078–50092. [CrossRef]
192. Yaokumah, W.; Wiafe, I. Analysis of machine learning techniques for anomaly-based intrusion detection. *Int. J. Distrib. Artif. Intell. (IJDAI)* **2020**, *12*, 20–38. [CrossRef]
193. Sadhu, P.K.; Yanambaka, V.P.; Abdelgawad, A. Internet of things: Security and solutions survey. *Sensors* **2022**, *22*, 7433. [CrossRef] [PubMed]
194. Sidhu, S.; Mohd, B.J.; Hayajneh, T. Hardware security in IoT devices with emphasis on hardware trojans. *J. Sens. Actuator Netw.* **2019**, *8*, 42. [CrossRef]
195. Hu, W.; Chang, C.H.; Sengupta, A.; Bhunia, S.; Kastner, R.; Li, H. An overview of hardware security and trust: Threats, countermeasures, and design tools. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2020**, *40*, 1010–1038. [CrossRef]
196. Meneghello, F.; Calore, M.; Zucchetto, D.; Polese, M.; Zanella, A. IoT: Internet of threats? A survey of practical security vulnerabilities in real IoT devices. *IEEE Internet Things J.* **2019**, *6*, 8182–8201. [CrossRef]
197. Yang, K.; Blaauw, D.; Sylvester, D. Hardware designs for security in ultra-low-power IoT systems: An overview and survey. *IEEE Micro* **2017**, *37*, 72–89. [CrossRef]
198. Cirne, A.; Sousa, P.R.; Resende, J.S.; Antunes, L. Hardware security for IoT identity assurance. *Development* **2021**, *9*, 11.
199. Pourrahmani, H.; Yavarinasab, A.; Monazzah, A.M.H.; Van herle, J. A review of the security vulnerabilities and countermeasures in the Internet of Things solutions: A bright future for the Blockchain. *Internet Things* **2023**, *23*, 100888. [CrossRef]
200. Jin, Y. Introduction to hardware security. *Electronics* **2015**, *4*, 763–784. [CrossRef]
201. Rahman, M.T.; Shi, Q.; Tajik, S.; Shen, H.; Woodard, D.L.; Tehranipoor, M.; Asadizanjani, N. Physical inspection & attacks: New frontier in hardware security. In Proceedings of the 2018 IEEE 3rd International Verification and Security Workshop (IVSW), Costa Brava, Spain, 2–4 July 2018; pp. 93–102.
202. Aqeel, M.; Ali, F.; Iqbal, M.W.; Rana, T.A.; Arif, M.; Auwul, M.R. A review of security and privacy concerns in the internet of things (IoT). *J. Sens.* **2022**, *2022*, 5724168. [CrossRef]
203. Japa, A.; Majumder, M.K.; Sahoo, S.K.; Vaddi, R.; Kaushik, B.K. Hardware security exploiting post-CMOS devices: Fundamental device characteristics, state-of-the-art countermeasures, challenges and roadmap. *IEEE Circuits Syst. Mag.* **2021**, *21*, 4–30. [CrossRef]
204. Li, K.F.; Attarmoghaddam, N. Challenges and methodologies of hardware security. In Proceedings of the 2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA), Krakow, Poland, 16–18 May 2018; pp. 928–933.
205. Tsvetanov, F. Sensor network hardware security. *AIP Conf. Proc.* **2022**, *2570*, 020019.
206. Batina, L.; Jauernig, P.; Mentens, N.; Sadeghi, A.R.; Stapf, E. INVITED: In Hardware We Trust: Gains and Pains of Hardware-assisted Security. In Proceedings of the 2019 56th ACM/IEEE Design Automation Conference (DAC), Las Vegas, NV, USA, 2–6 June 2019; pp. 1–4.
207. Hassija, V.; Chamola, V.; Gupta, V.; Jain, S.; Guizani, N. A survey on supply chain security: Application areas, security threats, and solution architectures. *IEEE Internet Things J.* **2020**, *8*, 6222–6246. [CrossRef]

208. Obeng, Y.; Nolan, C.; Brown, D. Hardware security through chain assurance. In Proceedings of the 2016 Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, Germany, 14–18 March 2016; pp. 1535–1537.
209. Akter, S.; Khalil, K.; Bayoumi, M. Hardware security in the internet of things: A survey. In Proceedings of the 2023 IEEE 36th International System-on-Chip Conference (SOCC), Santa Clara, CA, USA, 5–8 September 2023; pp. 1–6.
210. Maharmeh, H.A.; Alhawari, M.; Hung, C.C.; Ismail, M. Hardware security threats and countermeasures: A study of obfuscation, camouflaging and PUFs. *Int. J. Multimed. Intell. Secur.* **2019**, *3*, 271–292. [CrossRef]
211. Akter, S.; Khalil, K.; Bayoumi, M. A survey on hardware security: Current trends and challenges. *IEEE Access* **2023**, *11*, 77543–77565. [CrossRef]
212. Babaei, A.; Schiele, G. Physical unclonable functions in the internet of things: State of the art and open challenges. *Sensors* **2019**, *19*, 3208. [CrossRef]
213. Shariffuddin, S.; Sivamangai, N.; Napolean, A.; Naveenkumar, R.; Kamalnath, S.; Saranya, G. Review on Arbiter Physical Unclonable Function and its Implementation in FPGA for IoT Security Applications. In Proceedings of the 2022 6th International Conference on Devices, Circuits and Systems (ICDCS), Coimbatore, India, 21–22 April 2022; pp. 369–374. [CrossRef]
214. Al-Meer, A.; Al-Kuwari, S. Physical Unclonable Functions (PUF) for IoT Devices. *arXiv* **2022**, arXiv:2205.08587. [CrossRef]
215. Bergfalck, L.; Engström, J. *Designing a Physical Unclonable Function for Cryptographic Hardware*; Linköping University: Linköping, Sweden, 2021.
216. Liu, Y.; Xie, Y.; Bao, C.; Srivastava, A. A combined optimization-theoretic and side-channel approach for attacking strong physical unclonable functions. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2017**, *26*, 73–81. [CrossRef]
217. Shah, N.; Chatterjee, D.; Sapui, B.; Mukhopadhyay, D.; Basu, A. Introducing Recurrence in Strong PUFs for Enhanced Machine Learning Attack Resistance. *IEEE J. Emerg. Sel. Top. Circuits Syst.* **2021**, *11*, 319–332. [CrossRef]
218. Xu, Y.; Lao, Y.; Liu, W.; Zhang, Z.; You, X.; Zhang, C. Mathematical Modeling Analysis of Strong Physical Unclonable Functions. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2020**, *39*, 4426–4438. [CrossRef]
219. Ma, X.; Wang, P.; Li, G.; Zhou, Z. Machine learning attacks resistant strong PUF design utilizing response obfuscates challenge with lower hardware overhead. *Microelectron. J.* **2023**, *142*, 105977. [CrossRef]
220. Cao, Y.; Xu, J.; Wu, J.; Wu, S.; Huang, Z.; Zhang, K. Advances in Physical Unclonable Functions Based on New Technologies: A Comprehensive Review. *Mathematics* **2023**, *12*, 77. [CrossRef]
221. Su, H. Novel Design in Mixed-Signal and Machine Learning Resilient Architecture Physical Unclonable Functions. Ph.D. Thesis, University of Southampton, Southampton, UK, 2021.
222. Magyari, A.; Chen, Y. Integrating Lorenz Hyperchaotic Encryption with Ring Oscillator Physically Unclonable Functions (RO-PUFs) for High-Throughput Internet of Things (IoT) Applications. *Electronics* **2023**, *12*, 4929. [CrossRef]
223. Zhuang, Y.; Mursi, K.T.; Gaoxiang, L. A challenge obfuscating interface for arbiter PUF variants against machine learning attacks. *arXiv* **2021**, arXiv:2103.12935.
224. Avvaru, S.V.S.; Zhou, C.; Satapathy, S.; Lao, Y.; Kim, C.H.; Parhi, K.K. Estimating delay differences of arbiter PUFs using silicon data. In Proceedings of the 2016 Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, Germany, 14–18 March 2016; pp. 543–546.
225. Wang, K.; Shi, J.; Lai, W.; He, Q.; Xu, J.; Ni, Z.; Liu, X.; Pi, X.; Yang, D. All-silicon multidimensionally-encoded optical physical unclonable functions for integrated circuit anti-counterfeiting. *Nat. Commun.* **2024**, *15*, 3203. [CrossRef]
226. Chung, M.K.; Kim, M.U.; Han, J.W.; Yang, J.S.; Kim, B.J.; Jo, M.S.; Jung, S.Y.; Kim, S.H.; Yoon, J.B. Contribution of MEMS to Physical Unclonable Functions (PUFs): Random Configuration of PDMS Nano-Structure for Optical PUF. In Proceedings of the 2024 IEEE 37th International Conference on Micro Electro Mechanical Systems (MEMS), Austin, TX, USA, 21–25 January 2024; pp. 521–524. [CrossRef]
227. Lu, X.; Hong, L.; Sengupta, K. CMOS optical PUFs using noise-immune process-sensitive photonic crystals incorporating passive variations for robustness. *IEEE J. Solid-State Circuits* **2018**, *53*, 2709–2721. [CrossRef]
228. Yavuz, S.; Naroska, E.; Daniel, K. Vulnerabilities and Challenges in the Development of PUF-Based Authentication Protocols on FPGAs: A Brief Review. In Proceedings of the 2024 IEEE Conference on Dependable and Secure Computing (DSC), Tokyo, Japan, 6–8 November 2024; pp. 58–65. [CrossRef]
229. Yuyuan, L.; Xueqiang, S. A Highly Reliable Reconfigurable SRAM PUF Based on Error Correction Codes and Capacitor Preselection. *Authorea* 2024, *preprints*.
230. Rührmair, U. On the Security of PUF Protocols Under Bad PUFs and PUFs-Inside-PUFs Attacks. *Cryptology ePrint Archive, Paper 2016/322*. 2016. Available online: https://eprint.iacr.org/2016/322 (accessed on 11 April 2025).
231. Rührmair, U.; Sehnke, F.; Sölter, J.; Dror, G.; Devadas, S.; Schmidhuber, J. Modeling attacks on physical unclonable functions. In Proceedings of the 17th ACM conference on Computer and Communications Security, Chicago, IL, USA, 4–8 October 2010; pp. 237–249.
232. Zhang, J.; Shen, C. Set-based obfuscation for strong PUFs against machine learning attacks. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2020**, *68*, 288–300. [CrossRef]

233. Kroeger, T.; Cheng, W.; Guilley, S.; Danger, J.L.; Karimi, N. Assessment and Mitigation of Power Side-Channel-Based Cross-PUF Attacks on Arbiter-PUFs and Their Derivatives. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2022**, *30*, 187–200. [CrossRef]
234. Wang, Y.; Xi, X.; Orshansky, M. Lattice PUF: A strong physical unclonable function provably secure against machine learning attacks. In Proceedings of the 2020 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), San Jose, CA, USA, 7–11 December 2020; pp. 273–283.
235. Zhou, Z.; Li, G.; Wang, P. A challenge-screening strategy for enhancing the stability of strong PUF based on machine learning. *Microelectron. J.* **2023**, *131*, 105667. [CrossRef]
236. Wu, L.; Hu, Y.; Zhang, K.; Li, W.; Xu, X.; Chang, W. Flam-puf: A response–feedback-based lightweight anti-machine-learning-attack puf. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2022**, *41*, 4433–4444. [CrossRef]
237. Zhu, B.; Jiang, X.; Huang, K.; Yu, M. A Response-Feedback-Based Strong PUF with Improved Strict Avalanche Criterion and Reliability. *Sensors* **2023**, *24*, 93. [CrossRef] [PubMed]
238. Wang, Y.; Zhang, G.; Mei, X.; Gu, C. A High-Reliability, Non-CRP-Discard Arbiter PUF Based on Delay Difference Quantization. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2024**, *72*, 573–585. [CrossRef]
239. Wang, Y.; Wang, C.; Gu, C.; Cui, Y.; O'Neill, M.; Liu, W. A dynamically configurable PUF and dynamic matching authentication protocol. *IEEE Trans. Emerg. Top. Comput.* **2021**, *10*, 1091–1104. [CrossRef]
240. Zhou, Z.; Wang, P.; Li, G. Bagua Protocol: A Whole-Process Configurable Protocol for IoT Sensing Devices Security Based on Strong PUF. *IEEE Internet Things J.* **2024**, *11*, 805–819. [CrossRef]
241. Zhou, Z.; Li, G.; Wang, P.; Ye, M. Matrix encryption based anti-machine learning attack algorithm for strong PUF. In Proceedings of the 2021 IEEE 14th International Conference on ASIC (ASICON), Kunming, China, 26–29 October 2021; pp. 1–4.
242. Tun, N.W.; Mambo, M. Secure PUF-Based Authentication Systems. *Sensors* **2024**, *24*, 5295. [CrossRef]
243. Felicetti, C.; Lanuzza, M.; Rullo, A.; Saccà, D.; Crupi, F. Exploiting Silicon Fingerprint for Device Authentication Using CMOS-PUF and ECC. In Proceedings of the 2021 IEEE International Conference on Smart Internet of Things (SmartIoT), Jeju, Republic of Korea, 13–15 August 2021; pp. 229–236. [CrossRef]
244. Zhuang, Y.; Li, G. A lightweight PUF-based authentication protocol. *arXiv* **2024**, arXiv:2405.13146.
245. Jin, C.; Burleson, W.; van Dijk, M.; Rührmair, U. Programmable access-controlled and generic erasable PUF design and its applications. *J. Cryptogr. Eng.* **2022**, *12*, 413–432. [CrossRef]
246. Li, H.; Cao, W.; Wang, C.; Zhu, X.; Liao, G.; He, Z. FOM-CDS PUF: A Novel Configurable Dual State Strong PUF Based on Feedback Obfuscation Mechanism against Modeling Attacks. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **2023**, *106*, 1311–1321. [CrossRef]
247. Dubrova, E.; Näslund, O.; Degen, B.; Gawell, A.; Yu, Y. CRC-PUF: A machine learning attack resistant lightweight PUF construction. In Proceedings of the 2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), Stockholm, Sweden, 17–19 June 2019; pp. 264–271.
248. Zhang, Y.; He, Z.; Wan, M.; Liu, J.; Gu, H.; Zou, X. A SC PUF standard cell used for key generation and anti-invasive-attack protection. *IEEE Trans. Inf. Forensics Secur.* **2021**, *16*, 3958–3973. [CrossRef]
249. Lee, S.; Oh, M.K.; Kang, Y.; Choi, D. RC PUF: A low-cost and an easy-to-design PUF for resource-constrained IoT devices. In Proceedings of the Information Security Applications: 20th International Conference, WISA 2019, Jeju Island, Republic of Korea, 21–24 August 2019; Revised Selected Papers 20; Springer: Cham, Switzerland, 2020; pp. 275–285.
250. Wang, W.; Chen, Q.; Yin, Z.; Srivastava, G.; Gadekallu, T.R.; Alsolami, F.; Su, C. Blockchain and PUF-based lightweight authentication protocol for wireless medical sensor networks. *IEEE Internet Things J.* **2021**, *9*, 8883–8891. [CrossRef]
251. Asif, R.; Ghanem, K.; Irvine, J. Proof-of-puf enabled blockchain: Concurrent data and device security for internet-of-energy. *Sensors* **2020**, *21*, 28. [CrossRef] [PubMed]
252. Choi, S.; Zage, D.; Choe, Y.R.; Wasilow, B. Physically Unclonable Digital ID. In Proceedings of the 2015 IEEE International Conference on Mobile Services, New York City, NY, USA, 27 June–2 July 2015; pp. 105–111.
253. P, S.; Krishnammal, P.M. Study of different silicon Physical Unclonable Functions. In Proceedings of the 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, India, 23–25 March 2016; pp. 81–85. [CrossRef]
254. Joshi, S.; Mohanty, S.P.; Kougianos, E. Everything you wanted to know about PUFs. *IEEE Potentials* **2017**, *36*, 38–46. [CrossRef]
255. Ram, S.K.; Sahoo, S.R.; Das, B.B.; Mahapatra, K.; Mohanty, S.P. Securing Things: A Novel CRO Applicable in PUF and Recycled IC Detection. 2 June 2022, PREPRINT (Version 1). Available online: https://www.researchsquare.com/article/rs-1702083/v1 (accessed on 11 April 2025).
256. Bhargava, M. Reliable, Secure, Efficient Physical Unclonable Functions. Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 2013.
257. Schaub, A.; Danger, J.L.; Rioul, O.; Guilley, S. The big picture of delay-PUF dependability. In Proceedings of the 2020 European Conference on Circuit Theory and Design (ECCTD), Sofia, Bulgaria, 7–10 September 2020; pp. 1–4.

258. Jouini, Z.C.; Danger, J.L.; Bossuet, L. Performance evaluation of physically unclonable function by delay statistics. In Proceedings of the 2011 IEEE 9th International New Circuits and Systems Conference, Bordeaux, France, 26–29 June 2011; pp. 482–485.
259. Fedorkow, G.; Hardjono, T. Mind Your Roots of Trust! *Authorea* 2024, *preprints*.
260. CYSEC. Securing Edge Devices: The Crucial Role of Root of Trust in a Connected World. Available online: https://www.cysec.com/iot-edge/ (accessed on 30 October 2024).
261. Knowledge Center, S.E. Root of Trust: Trusted Environment for Secure Functions. Available online: https://semiengineering.com/knowledge_centers/semiconductor-security/root-of-trust/ (accessed on 30 October 2024).
262. Zimmer, V.; Krau, M. Establishing the Root of Trust. August 2016. Available online: https://www.researchgate.net/profile/Vincent-Zimmer-5/publication/307478971_Establishing_the_Root_of_Trust/links/668c7406c1cf0d77ffc3837b/Establishing-the-Root-of-Trust.pdf (accessed on 30 October 2024).
263. Ehret, A.; Moore, P.; Stojkov, M.; Kinsy, M.A. Hardware Root-of-Trust Support for Operational Technology Cybersecurity in Critical Infrastructures. In Proceedings of the 2023 IEEE High Performance Extreme Computing Conference (HPEC), Boston, MA, USA, 25–29 September 2023; pp. 1–7.
264. Chaintoutis, C.; Akriotou, M.; Mesaritakis, C.; Komnios, I.; Karamitros, D.; Fragkos, A.; Syvridis, D. Optical PUFs as physical root of trust for blockchain-driven applications. *IET Softw.* **2019**, *13*, 182–186. [CrossRef]
265. Rojas-Muñoz, L.F.; Sánchez-Solano, S.; Martínez-Rodríguez, M.C.; Camacho-Ruiz, E.; Navarro-Torrero, P.; Karmakar, A.; Fernández-García, C.; Tena-Sánchez, E.; Potestad-Ordóñez, F.E.; Casado-Galán, A.; et al. Cryptographic Security Through a Hardware Root of Trust. In Proceedings of the International Symposium on Applied Reconfigurable Computing, Aveiro, Portugal, 20–22 March 2024; pp. 106–119.
266. Chuang, K.K.H.; Chen, H.M.; Wu, M.Y.; Yang, E.C.S.; Hsu, C.C.H. Quantum Tunneling PUF: A Chip Fingerprint for Hardware Security. In Proceedings of the 2021 International Symposium on VLSI Technology, Systems and Applications (VLSI-TSA), Hsinchu, Taiwan, 19–22 April 2021; pp. 1–2. [CrossRef]
267. Hamadeh, H.; Tyagi, A. Physical unclonable functions (pufs) entangled trusted computing base. In Proceedings of the 2019 IEEE International Symposium on Smart Electronic Systems (iSES)(Formerly iNiS), Rourkela, India, 16–18 December 2019; pp. 177–180.
268. Bathalapalli, V.K.; Mohanty, S.P.; Kougianos, E.; Iyer, V.; Rout, B. PUFchain 4.0: Integrating PUF-based TPM in distributed ledger for security-by-design of IoT. In Proceedings of the Great Lakes Symposium on VLSI 2023, Knoxville, TN, USA, 5–7 June 2023; pp. 231–236.
269. Colombier, B.; Bossuet, L.; Fischer, V.; Hély, D. Key reconciliation protocols for error correction of silicon PUF responses. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 1988–2002. [CrossRef]
270. Schrijen, G.J.; Garlati, C. Physical unclonable functions to the rescue. In Proceedings of the Embedded World, Nuremberg, Germany, 27 February–1 March 2018. [CrossRef]
271. Dave McCarthy, P.R.; Kanaracus, C. *Increasing Intelligence at the Edge with AI*; Technical Report; International Data Corporation (IDC): Needham, MA, USA, 2022.
272. Ahmad, J.; Jervis, M.; Venkata, R. *Intel® FPGAs and SoCs with Intel® FPGA AI Suite and OpenVINO Toolkit Drive Embedded/Edge AI/Machine Learning Applications*; Infotech: Arlington, VA, USA, 2022.
273. Xu, C.; Jiang, S.; Luo, G.; Sun, G.; An, N.; Huang, G.; Liu, X. The Case for FPGA-Based Edge Computing. *IEEE Trans. Mob. Comput.* **2022**, *21*, 2610–2619. [CrossRef]
274. Al-Asli, M.; Elrabaa, M.E.; Abu-Amara, M. FPGA-based symmetric re-encryption scheme to secure data processing for cloud-integrated internet of things. *IEEE Internet Things J.* **2018**, *6*, 446–457. [CrossRef]
275. Martínez-Rodríguez, M.C.; Rojas-Muñoz, L.F.; Camacho-Ruiz, E.; Sánchez-Solano, S.; Brox, P. Efficient RO-PUF for generation of identifiers and keys in resource-constrained embedded systems. *Cryptography* **2022**, *6*, 51. [CrossRef]
276. Mukhtar, N.; Mehrabi, A.; Kong, Y.; Anjum, A. Edge enhanced deep learning system for IoT edge device security analytics. *Concurr. Comput. Pract. Exp.* **2021**, *35*, e6764. [CrossRef]
277. Huffmire, T.; Brotherton, B.; Sherwood, T.; Kastner, R.; Levin, T.; Nguyen, T.D.; Irvine, C. Managing security in FPGA-based embedded systems. *IEEE Des. Test Comput.* **2008**, *25*, 590–598. [CrossRef]
278. Kolosov, D.; Kelefouras, V.; Kourtessis, P.; Mporas, I. Anatomy of Deep Learning Image Classification and Object Detection on Commercial Edge Devices: A Case Study on Face Mask Detection. *IEEE Access* **2022**, *10*, 109167–109186. [CrossRef]
279. *Vitis AI Library User Guide—UG1354 (v2.0)*; AMD: Santa Clara, CA, USA, 2022; pp. 1–401.
280. Rahman, A.; Hassanain, E.; Hossain, M.S. Towards a secure mobile edge computing framework for Hajj. *IEEE Access* **2017**, *5*, 11768–11781. [CrossRef]
281. Cao, K.; Liu, Y.; Meng, G.; Sun, Q. An overview on edge computing research. *IEEE Access* **2020**, *8*, 85714–85728. [CrossRef]
282. Williams, M.; Emeteveke, I.; Adeyeye, O.J.; Emehin, O. Enhancing Data Forensics through Edge Computing in IoT Environments. *Int. J. Res. Publ. Rev.* **2024**, *5*, 2970–2985. [CrossRef]
283. Ming, Y.; Wang, C.; Liu, H.; Zhao, Y.; Feng, J.; Zhang, N.; Shi, W. Blockchain-Enabled Efficient Dynamic Cross-Domain Deduplication in Edge Computing. *IEEE Internet Things J.* **2022**, *9*, 15639–15656. [CrossRef]

284. Wang, Y.; Yang, X. Research on Edge Computing and Cloud Collaborative Resource Scheduling Optimization Based on Deep Reinforcement Learning. *arXiv* **2025**, arXiv:2502.18773.
285. Luo, Q.; Hu, S.; Li, C.; Li, G.; Shi, W. Resource scheduling in edge computing: A survey. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 2131–2165. [CrossRef]
286. Yang, Q.; Liu, Y.; Chen, T.; Tong, Y. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol. (TIST)* **2019**, *10*, 1–19. [CrossRef]
287. Cui, Z.; Zhao, P.; Hu, Z.; Cai, X.; Zhang, W.; Chen, J. An improved matrix factorization based model for many-objective optimization recommendation. *Inf. Sci.* **2021**, *579*, 1–14. [CrossRef]
288. Zhai, K.; Ren, Q.; Wang, J.; Yan, C. Byzantine-robust federated learning via credibility assessment on non-IID data. *arXiv* **2021**, arXiv:2109.02396. [CrossRef] [PubMed]