The State of Edge AI

Advait Jayant[1, 3], Matthew Sheldon[2, 3], Sungjung Kim[2, 3], and Swastik Shrivastava[3]

[1]London Business School, The Regent's Park, London, NW1 4SA

[2]Imperial College London, South Kensington, London SW7 2AZ, UK

[3]Periphery (perilabs.net)

[*]Contact: peri@perilabs.net

8th October 2024

Abstract

Edge Artificial Intelligence (Edge AI) is emerging as a crucial solution to the escalating limitations of traditional cloud-based AI, including high latency, privacy concerns, and substantial bandwidth and computational resource demands. This report explores the necessity for Edge AI in the context of increasingly large and complex AI models, highlighting how decentralized processing on devices such as smartphones, IoT gadgets, and embedded systems can enable real-time data analysis, enhance data privacy, and optimize resource utilization. The report presents a long-term thesis that Edge AI will become integral to the future of AI across diverse industries, driving innovations in healthcare, robotics, virtual assistants, and autonomous driving by providing localized intelligence and reducing dependency on centralized cloud infrastructures.

Keywords: Edge AI, Cloud Computing, Decentralisation, Crypto, Large Language Models (LLMs).

Acknowledgements

We give special thanks to JJ Agcaoili, Yeona Kim and Rayaan Sheikh and Sooyeon Hong for late nights and unparalleled teamwork. We thank Sven Wellmann (Polychain) for providing endless support and feedback through conversations, comments, ideas, and advice. We thank Luca Prosperi ($M\hat{}0$) for encouraging us towards building a long-term thesis for the space. We give special thanks to the BeWater team for translating this document into Chinese and amplifying our reach. We thank the following people for their feedback and discussions that shaped our thesis: Bowen Li (EigenLayer and Apple), Paul Taylor (BlackRock), Sriram Vishwanath (UT Austin), Josie Leung (MilkyWay), Joy Song (ABCDE), Mike Arpaia (Moonfire), Geoff YuHasker (Electric Capital), Bhavin Vaid and Daniel Howard (Halo Capital), Richard Muirhead (Fabric Ventures), Altan Tutar and Sam Wang (Nuffle Labs), Jerry Xu (Stellar Foundation), Joyce Chin (Animoca Brands), Nirav Murthy (Camp Network), Young Ling, Chris Pizzo (Druid Ventures), Aly Madhavji (Blockchain Founders Fund), Clemens Scherf (Lucid), Nelson Paul (Pivot Accelerator), Yarco Hayduk (Pragma), Dan Patterson (Sfermion) and Alex Gedeveni (Sfermion).

We give special thanks to Yarco Hayduk (Pragma), Kevin Zhang (Lightspark), Young Ling (Founder, Stealth), Aaron (Crynux), Sven Wellmann (Polychain), and Marko Stokić (Oasis Network) for providing us with detailed feedback.

Finally, we thank everyone who helped review the report including Jerry Xu (Stellar Foundation), Dylan Zhang (Pond) and Bill Shi (Pond), Geoff YuHasker (Electric Capital), Whitney Gibbs (Ringfence), Don Gossen (Nevermined), Prashant Maurya (Spheron), Chris Pizzo (Druid Ventures), Calanthia Mei (Masa Network), Paul Taylor (BlackRock), Chloe and LuLu (BeWater), Jiahao Sun (FLock), Carlos Castellanos (Samsung Next), Oliver Jaros (CMT Digital), Anna Kazlauskas (Vana), Storm (Anagram), Lucas Tcheyan (Galaxy Digital), and Caleb Shack (Big Brain).

Contents

0.1. Foreword

0.1.1 Note from the Authors

Edge AI is

Chapter 1

The Need for Edge AI

1.1. Introduction

1.1.1 The Rise of AI

Artificial intelligence (AI) has rapidly transitioned from a theoretical concept to a practical technology that permeates various aspects of modern life. This evolution is driven by advancements in machine learning algorithms, increased computational power, and the availability of large datasets. We foresee that AI will be instrumental in solving the world's most complex problems, leading to breakthroughs across multiple industries.

Breakthroughs in AI Applications

AI has achieved remarkable successes in several fields:

- **Healthcare:** Deep learning models assist in early disease detection, personalized treatment plans, and drug discovery. For example, AI algorithms have been developed to detect diabetic retinopathy from retinal images with high accuracy, potentially preventing blindness through early intervention (Gulshan et al., 2016).

- **Transportation:** Autonomous vehicles leverage AI for navigation, obstacle detection, and decision-making processes. Companies like Tesla and Waymo use

machine learning algorithms to interpret sensor data in real-time, enhancing road safety and efficiency (Badue et al., 2021b).

- **Code Generation:** AI tools improve developer productivity by automating code completion, bug detection, and code optimization. GitHub's Copilot, powered by OpenAI's Codex, can suggest code snippets and functions based on context, streamlining the development process (M. Chen, Tworek, Jun, Yuan, Pinto, Kaplan, Zaremba et al., 2021).

- **Arts:** Generative AI models create original music, art, and literature. Projects like OpenAI's DALL · E can generate images from textual descriptions, pushing the boundaries of creativity and expanding the role of machines in artistic expression (Ramesh et al., 2021).

The Emergence of Large Language Models (LLMs)

A significant milestone in AI is the development of Large Language Models (LLMs), such as OpenAI's GPT-3 and GPT-4. These models are trained on vast amounts of text data, allowing them to understand context, generate human-like text, and perform a variety of language tasks Thompson et al., 2020. LLMs have completely altered natural language processing by:

- **Enhancing Conversational Agents:** LLMs power sophisticated chatbots capable of engaging in coherent and context-aware dialogues, improving customer service and user experience.

- **Facilitating Code Generation:** Models like Codex translate natural language descriptions into functional code, aiding in software development and learning (M. Chen, Tworek, Jun, Yuan, Pinto, Kaplan et al., 2021)

- **Enabling Advanced Translation and Summarization:** LLMs provide high-quality translations and concise summaries, breaking language barriers and distilling information efficiently.

However, these advancements come with challenges. As models grow in size and complexity, they require more computational resources and specialized hardware, leading to increased latency and energy consumption. For instance, GPT-3.5 has an average response time of 35 milliseconds per token, while the more advanced GPT-4 has a response time of 94 milliseconds per token (OpenAI, 2023). This latency can hinder real-time applications and negatively impact user experience.

Moreover, the growth rate of AI models is outpacing the improvements in hardware capabilities described by Moore's Law, creating a gap between AI demand and computing supply (Thompson et al., 2020).

1.1.2 Challenges with Traditional AI Deployments

As AI models become more sophisticated, they present significant challenges in terms of deployment and scalability. Traditional AI deployments, particularly those relying on cloud-based infrastructures, are increasingly strained under the demands of modern AI applications.

Computational Resource Demands

The surge in AI capabilities is closely tied to the exponential growth in model size and complexity. Large Language Models (LLMs) like GPT-3 and GPT-4 consist of billions of parameters, requiring immense computational resources for both training and inference (Brown et al., 2020). Training these models demands specialized hardware such as Graphics Processing Units (GPUs) and Tensor Processing Units (TPUs), which are expensive and consume substantial energy (Jouppi et al., 2017a). Moreover, deploying these models for real-time applications poses additional challenges. Inference—using the trained model to make predictions—can be resource-intensive. For instance, running GPT-3 in a production environment requires significant memory and processing power to achieve acceptable latency levels (Patterson et al., 2021a). This high resource demand limits the accessibility of advanced AI technologies to organizations with substantial infrastructure capabilities. The computational intensity

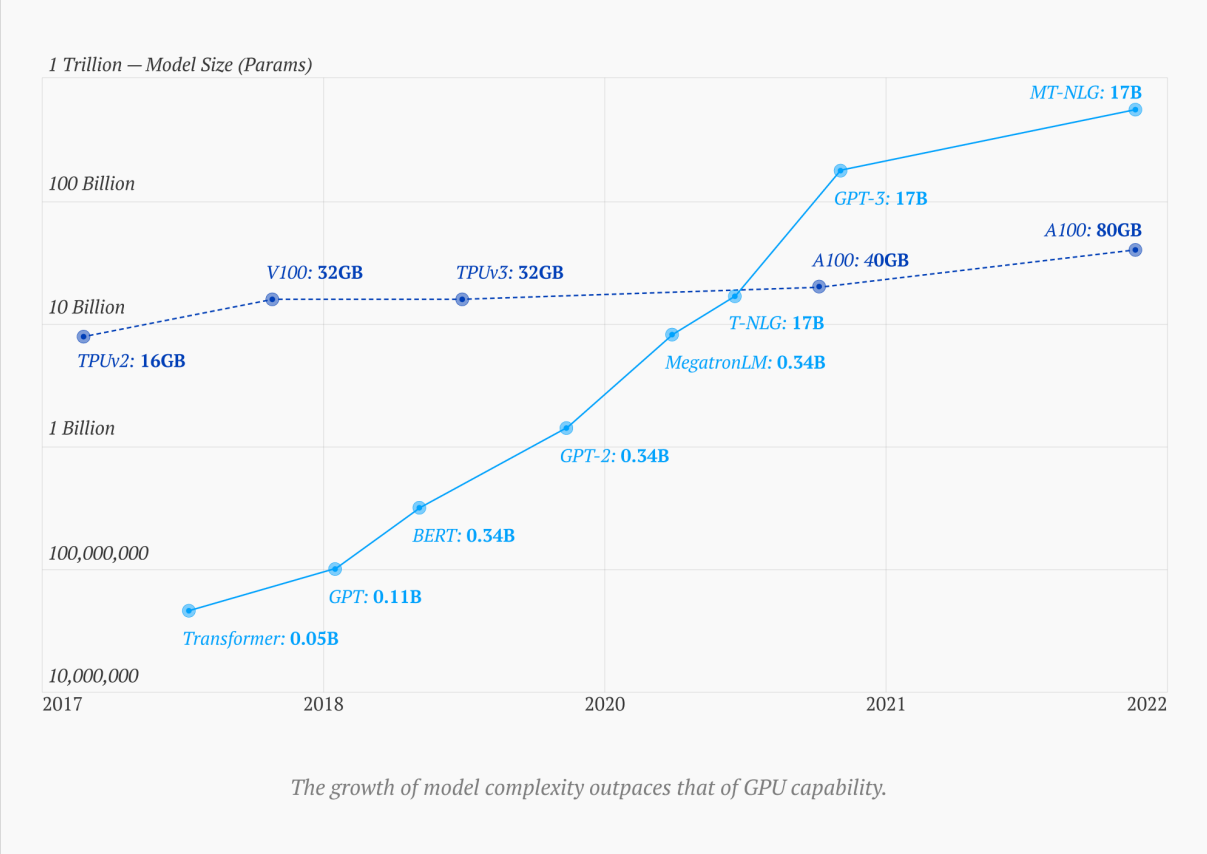

Figure 1.1: The growth of AI model size exceeds the growth in GPU memory (Thompson et al., 2020).

also has environmental implications. Data centers housing the hardware for AI computations consume large amounts of electricity, contributing to carbon emissions. A study highlighted that training a single AI model can emit nearly as much carbon as five cars in their lifetimes (Strubell et al., 2019).

The Growing Gap Beyond Moore's Law

Moore's Law, the observation that the number of transistors on a microchip doubles approximately every two years, has historically predicted the exponential growth of computing power (Moore, 1965). However, the rate of AI model growth is outpacing these hardware improvements, leading to a widening gap between computational demand and supply (Thompson et al., 2020). Recent AI models have increased in size by orders of magnitude within a few years. This rapid expansion exceeds the incremental improvements in hardware capabilities, creating a bottleneck for AI development. This disparity necessitates a co-design strategy, where both hardware and software are op-

timized in tandem to meet the escalating demands (Sze, Chen et al., 2017a). Without innovative approaches to bridge this gap, the progression of AI may be hindered by physical and economic limitations. The growing computational demands also affect latency and user experience. High model complexity can result in slower response times, which is detrimental for applications requiring real-time interaction (Patterson et al., 2021a). Users are sensitive to delays; studies show that even a one-second increase in response time can lead to a significant drop in user engagement (Nah, 2004). Beyond latency and user experience, current AI models are restrictive, offering users limited flexibility to optimize or fine-tune them according to their individual needs. These challenges underscore the need for alternative deployment strategies, such as Edge AI, which aims to process data closer to the source using optimized models and hardware to reduce latency and resource consumption.

1.2. Limitations of Cloud-Based AI

Cloud-based AI services have played a pivotal role in improving access to advanced AI capabilities. However, they present several limitations that can hinder their effectiveness in certain applications. Key among these limitations are latency issues, privacy concerns, bandwidth and cost implications.

1.2.1 Latency Issues

Latency, the delay between a user's action and the system's response, is a critical factor in the performance of AI applications. High latency can degrade user experience and limit the applicability of AI in time-sensitive contexts.

Response Times in Cloud AI Services

GPT-3.5 and GPT-4 Latencies Large Language Models (LLMs) like GPT-3.5 and GPT-4 require substantial computational resources due to their massive sizes. Running these models in the cloud introduces not only computational latency but

also network latency due to data transmission times.

Impact on User Experience High latency adversely affects user experience, especially in interactive applications where responsiveness is crucial. Users begin to perceive delays at around 100 milliseconds, and delays exceeding 1 second can disrupt the flow of interaction (Nielsen, 1994). In conversational AI applications, such as chatbots or virtual assistants, latency can make interactions feel sluggish and unnatural, reducing user engagement and satisfaction. Moreover, in critical applications like emergency response systems or financial trading platforms, delays can have significant consequences, leading to losses or safety risks.

Real-Time Application Constraints

Case Study: Google Assistant to Meta's Real-Time Translation Google Assistant initially relied heavily on cloud processing for speech recognition and natural language understanding, resulting in latency due to data transmission and remote computation (Schuster et al., 2020b). To enhance responsiveness, Google shifted to on-device processing by deploying compressed versions of their neural networks directly on user devices. This transition reduced latency from hundreds of milliseconds to under 500 milliseconds, significantly improving user experience (Schuster et al., 2020b). By processing voice commands locally, Google Assistant can respond more quickly, making interactions smoother and more natural. Apple has made significant strides in improving Siri's responsiveness by implementing on-device processing. Tesla's self-driving technology requires real-time processing of vast amounts of sensor data. They've developed custom AI chips to handle this processing on-board, reducing latency in decision-making for their autonomous driving features.(Tesla, 2019) Meta has been working on real-time translation for its messaging platforms, which requires low-latency natural language processing to provide seamless communication across language barriers (A. Fan et al., 2021).

User Sensitivity to Delays According to Nielsen (Nielsen, 1994), delays of more than 1 second interrupt a user's thought process, and delays over 10 seconds can make users abandon tasks altogether. In applications like online gaming, virtual reality, or autonomous driving, latency is not just an inconvenience but a critical factor that can affect functionality and safety (Singh & Sharma, 2019). Therefore, reducing latency is essential for the effectiveness and adoption of AI technologies in real-time applications.

1.2.2 Limited Model Portability

The portability of AI models is a significant concern, particularly for applications that require flexibility, customization, or deployment on diverse hardware. Many advanced AI models are typically hosted in centralized data centers, which restricts end-users' ability to access, customize, or optimize them for their specific needs. This limitation creates challenges in deploying AI solutions across different environments.

Hardware Constraints and Accessibility Most state-of-the-art AI models are large and resource-intensive, requiring substantial computational power, such as GPUs or TPUs, to operate efficiently. This means they are often inaccessible to users with standard hardware or smaller devices like smartphones or IoT devices. As a result, many AI solutions remain confined to cloud infrastructure, limiting their application in real-time or offline scenarios where cloud connectivity is not reliable or practical. This dependence on high-end hardware hampers the wider adoption of AI in diverse fields, especially where low-latency, on-device processing is essential.

Lack of Customization and Fine-Tuning The centralized nature of AI models often results in a "one-size-fits-all" solution, offering limited opportunities for customization. End-users typically have little or no control over model parameters, preventing them from adapting the model to suit their unique requirements or use cases. For example, a financial institution may fine-tune a model to detect fraud patterns specific to its regional market. Due to the centralized model's inability to adapt to these unique, localized transaction behaviors, the institution might experience less effective

fraud detection, potentially resulting in increased false positives or missed fraudulent activities, thereby necessitating a completely different solution tailored to their needs.

Dependence on Cloud Infrastructure The reliance on cloud infrastructure for running and accessing AI models poses significant challenges in terms of latency, connectivity, and operational costs. In environments with poor internet connectivity or in applications requiring immediate, real-time responses (e.g., autonomous vehicles or robotics), the dependence on centralized cloud models introduces delays that can be detrimental. Additionally, frequent data transfers to and from the cloud can be costly, making it impractical for some users or organizations to maintain continuous access to these models.

Regulatory and Data Sovereignty Challenges Deploying AI models through centralized data centers can raise regulatory concerns, especially when operating across different jurisdictions with varying data sovereignty laws. In cases where data must remain within a specific geographic region or be processed on local devices, the lack of model portability makes compliance difficult. For organizations operating in sectors with strict regulatory requirements, such as finance or healthcare, this limitation can impede the adoption of AI technologies. By addressing these issues, solutions such as model distillation, edge AI, and federated learning offer pathways to enhance model portability, allowing more adaptable and efficient use of AI across a wider range of devices and environments.

1.2.3 Privacy Concerns

Privacy is a paramount concern in AI applications, particularly when handling sensitive user data. Cloud-based AI services often require transmitting and storing personal data on remote servers, raising several privacy issues.

Risks of Data Breaches Storing data in the cloud exposes it to potential cyberattacks and data breaches. High-profile incidents have demonstrated that even well-

secured cloud services are vulnerable. For example, the 2019 Capital One data breach affected over 100 million customers due to a misconfigured firewall (U.S. Department of Justice, 2020). Data breaches can lead to unauthorized access to sensitive information, including personal identifiers, financial details, and private communications. In AI applications, this could mean exposure of confidential information with virtual assistants or sensitive personal data processed by AI services.

Opaque Data Policies and User Trust Cloud AI providers often have complex and opaque data policies that users may not fully understand. This lack of transparency can erode user trust (Martin, 2019). Users may be unaware of how their data is collected, used, or shared with third parties. Additionally, some AI services may use personal data to improve their models without explicit user consent. Trust is critical for the widespread adoption of AI technologies. If users are uncertain about how their data is handled, they may be reluctant to use cloud-based AI services, especially for applications involving sensitive information.

Regulatory Compliance (e.g., GDPR) Regulations like the General Data Protection Regulation (GDPR) in the European Union impose strict requirements on data privacy and protection (European Parliament and Council of European Union, 2016a). GDPR mandates that personal data must be processed lawfully, transparently, and for specified legitimate purposes. Cloud-based AI services must navigate complex regulatory landscapes, ensuring compliance across different jurisdictions. Non-compliance can result in severe penalties, including fines of up to 4 percent of annual global turnover or €20 million, whichever is greater (European Parliament and Council of European Union, 2016a). Furthermore, GDPR gives individuals rights over their data, such as the right to access, correct, and delete personal information. Implementing these rights in cloud environments can be challenging, particularly when data is distributed across multiple servers and locations.

1.2.4 Bandwidth Costs

The reliance on cloud infrastructure for AI services entails significant bandwidth usage and associated costs, affecting both service providers and users.

High Data Transfer Costs Transferring large amounts of data to and from cloud servers consumes substantial bandwidth, leading to high operational costs (Cisco, 2020b). Applications like video analytics, autonomous vehicles, and Internet of Things (IoT) devices generate massive data streams that need to be processed. For example, autonomous vehicles can produce up to 4 terabytes of data per day (Intel, 2016). Transmitting this volume of data to the cloud for processing is not only impractical but also prohibitively expensive. High data transfer costs can limit the scalability of cloud-based AI solutions.

Infrastructure Scaling Costs As the number of users grows, cloud-based AI services face scalability challenges. Increased demand requires proportional expansion of cloud infrastructure, including servers, storage, and networking capabilities (Armbrust et al., 2010a). Scaling up infrastructure is capital-intensive and may introduce additional latency due to network congestion. Moreover, service providers may pass these increased costs onto users, potentially making AI services less affordable. In regions with limited network infrastructure, bandwidth limitations can further hinder scalability and accessibility.

Environmental Impact of Data Centers Data centers consume significant amounts of energy, contributing to environmental concerns (Jones, 2018). The energy consumption of data centers globally accounts for about 1 percent of the world's electricity usage, with projections indicating a rising trend (International Energy Agency, 2021). The environmental footprint is exacerbated by the energy-intensive nature of AI computations, particularly for training and running large models. This raises sustainability issues and pressures companies to adopt greener practices, which can be costly. Additionally, the environmental impact may influence public perception and

regulatory policies, potentially affecting the operation of cloud-based AI services.

1.3. Edge AI: Bringing Intelligence Closer

1.3.1 Definition and Overview of Edge AI

Contrast with Cloud Computing

In traditional cloud computing, data generated by end devices is transmitted over the internet to centralized servers for processing and analysis. This approach has limitations, including latency due to network transmission times, potential privacy risks from data exposure, and dependence on stable internet connectivity (Satyanarayanan, 2017a). Edge AI addresses these issues by performing computations on or near the data source. This reduces the amount of data that needs to be sent to the cloud, lowering latency and bandwidth usage while enhancing privacy and security. While cloud computing offers scalable resources and centralized management, Edge AI provides faster response times and localized intelligence (X. Xu et al., 2018).

Evolution of Edge Computing

Edge computing has evolved with advancements in hardware miniaturization, increased computational power of edge devices, and the need for real-time data processing (Porambage et al., 2018). The proliferation of IoT devices generating massive amounts of data necessitated a shift from centralized processing to distributed architectures. Technological innovations, such as specialized edge processors and efficient AI algorithms, have enabled complex computations to be performed on resource-constrained devices. This evolution is driven by applications requiring low latency, such as autonomous vehicles, smart manufacturing, and real-time health monitoring (Mach & Becvar, 2017a).

1.3.2 Benefits of Edge AI

Edge AI offers several advantages over traditional cloud-based AI deployments, addressing key challenges related to latency, privacy, bandwidth, personalization, and reliability.

Reduced Latency and Faster Response Times

By processing data locally, Edge AI significantly reduces the time it takes for data to travel to a central server and back, thereby decreasing latency (T. Chen et al., 2019). This is crucial for applications requiring immediate responses, such as autonomous driving, where delays can have serious safety implications. For example, in autonomous vehicles, decision-making processes need to occur within milliseconds to react appropriately to dynamic driving conditions (B. Li, Li & Liu, 2018a). Edge AI enables rapid data analysis and action without the delays associated with cloud communication.

Improved Model Portability and Deployment

Techniques such as model distillation, quantization, and Edge AI significantly enhance model portability by reducing the size and computational requirements of AI models, enabling them to run efficiently on smaller devices (S. Wang et al., 2017). This approach allows users to deploy AI models across a wider range of hardware, from smartphones to IoT devices, without relying on centralized data centers. By making models more lightweight and adaptable, these techniques facilitate customization and fine-tuning to suit specific applications, allowing organizations to tailor AI solutions to their unique needs. This improved portability not only broadens the accessibility of AI but also reduces dependence on cloud infrastructure, leading to lower latency, increased responsiveness, and reduced operational costs.

Enhanced Privacy and Data Security

Edge AI enhances privacy by keeping sensitive data on the local device rather than transmitting it over networks to cloud servers, where it could be vulnerable to interception or breaches (J. Ren et al., 2019). This is particularly important in applications handling personal or confidential information, such as health data or financial transactions. Processing data locally also aligns with data protection regulations like the General Data Protection Regulation (GDPR), which emphasizes user consent and data minimization (European Parliament and Council of European Union, 2016b).

Lower Bandwidth Usage and Cost Savings

By reducing the need to transmit large volumes of data to and from cloud servers, Edge AI decreases bandwidth consumption (S. Wang et al., 2017). This leads to cost savings for both service providers and users, particularly in scenarios with limited network infrastructure or high data transmission costs. For instance, IoT devices generating continuous streams of data can offload processing to the edge, minimizing network congestion and associated expenses (J. Lin et al., 2017a).

Improved Personalization and Localized Processing

Edge AI allows for more personalized experiences by leveraging data that remains on the user's device (Lane et al., 2015). Applications can adapt to individual user behaviors and preferences without compromising privacy. In the context of virtual assistants, on-device processing enables the assistant to learn from the user's interactions and usage patterns, providing more relevant responses and recommendations (Kugler, 2018).

Reliability and Offline Capabilities

Edge AI enhances system reliability by reducing dependence on constant internet connectivity (Premsankar et al., 2018b). Applications can continue functioning even when network connections are unstable or unavailable. For example, in remote or

rural areas with limited connectivity, Edge AI enables critical services like medical diagnostics or agricultural monitoring to operate effectively without relying on cloud servers (Sood & Mahajan, 2017).

1.3.3 Case Studies Showcasing Advantages of Edge AI

Face Recognition Systems

Implementing face recognition algorithms at the edge has shown significant improvements in performance and privacy. Edge-based face recognition systems process images locally, reducing latency and eliminating the need to transmit sensitive biometric data over networks (Z. Yang & Yu, 2017). A study comparing cloud-based and edge-based face recognition found that edge computing reduced response times by 2.5x to 4.5x, depending on the cloud service (N. Zhang et al., 2018). This improvement is critical for applications like security systems or mobile authentication, where quick and secure verification is essential.

Real-Time Analytics in IoT Devices

IoT devices equipped with Edge AI capabilities can perform real-time analytics, enabling immediate decision-making without cloud dependence (S. Deng et al., 2020a). For instance, industrial sensors can detect anomalies and trigger preventive actions instantly, enhancing operational efficiency and safety. In smart homes, edge-enabled devices can adjust environmental controls based on real-time occupancy and usage patterns, optimizing energy consumption and improving user comfort (Alaa et al., 2017).

Wearables for Health Monitoring

Edge AI in wearable devices is revolutionizing personal health monitoring by enabling real-time analysis of vital signs and activity patterns. This local processing reduces latency and enhances privacy, which is crucial for handling sensitive health data. For example, edge-enabled watches can improve the speed of AFib detection by processing

data directly on the device, rather than sending it to a data center for analysis. By keeping sensitive health data on the device, edge computing enhances user privacy and data security. Another application is in continuous glucose monitoring for diabetes patients. Edge-enabled devices perform more frequent or continuous checks and interventions without relying on constant cloud connectivity.

Autonomous Vehicles and Drones

Edge computing significantly enhances the capabilities of autonomous vehicles and drones by enabling real-time decision-making, which is crucial for navigation and safety. By processing data locally, edge computing eliminates the delays that come with sending data to and receiving instructions from a cloud server. This real-time processing is especially vital in urban environments, where conditions change rapidly, and quick responses are needed to avoid collisions.

In dense urban areas, signal interference and network congestion are common challenges. Edge processing ensures that drones operate safely even when cloud connectivity is limited or disrupted. By reducing the need for constant high-bandwidth communication with cloud servers, edge computing becomes more reliable in such environments. This advantage is particularly beneficial for applications like package delivery and search-and-rescue operations. Edge processing keeps sensitive data, such as visual information of the environment, local, minimizing privacy concerns and security risks associated with transmitting data to the cloud. It also allows systems to be fine-tuned to specific urban settings, improving performance in navigating local obstacles. The reduced dependence on cloud communication lowers power consumption, potentially extending flight times. In agricultural applications, edge AI enables real-time crop analysis and targeted pesticide application, which reduces pesticide usage while maintaining crop yields, offering both economic and environmental advantages.

1.4. Technological Advancements Enabling AI on the Edge

Advancements in both hardware and software technologies have been pivotal in making Edge AI a practical and efficient solution. Innovations in specialized hardware components, algorithm optimization techniques, and co-design strategies have collectively addressed the challenges of deploying AI models on resource-constrained edge devices.

1.4.1 Hardware Innovations

The development of specialized hardware has significantly enhanced the computational capabilities of edge devices, enabling them to run complex AI algorithms efficiently.

Specialized Edge AI Chips (e.g., Edge TPUs)

Specialized chips designed for AI computations at the edge, such as Google's Edge Tensor Processing Units (Edge TPUs), have revolutionized edge computing. Edge TPUs are application-specific integrated circuits (ASICs) optimized for running deep learning models, particularly convolutional neural networks (CNNs) (Jouppi et al., 2017b). These chips offer high performance with low power consumption, making them ideal for real-time inference on edge devices. Edge TPUs accelerate AI workloads by performing matrix multiplications and convolutions efficiently. They are designed to handle integer quantized models, which are smaller and faster than their floating-point counterparts (Google Cloud, 2019). By offloading AI tasks to dedicated hardware, edge devices can achieve higher throughput and lower latency.

Mobile Processors with Integrated NPUs

Mobile processors now often include integrated Neural Processing Units (NPUs), which are specialized co-processors designed to accelerate machine learning tasks. Companies like Apple, Huawei, and Qualcomm have developed processors with built-in NPUs

to enhance on-device AI capabilities (L. Deng et al., 2020a). For example, Apple's A14 Bionic chip includes a 16-core Neural Engine capable of performing 11 trillion operations per second (Apple Inc., 2020c). Similarly, Huawei's Kirin series integrates NPUs that support mixed-precision computations, improving performance and energy efficiency (Huawei, 2019). These NPUs enable smartphones and tablets to run advanced AI applications, such as real-time image processing and natural language understanding, without relying on cloud services.

Energy-Efficient Computing Platforms

Energy efficiency is crucial for battery-powered edge devices. Advances in energy-efficient computing platforms, such as ARM's big.LITTLE architecture and NVIDIA's Jetson Nano, have enabled edge devices to perform complex AI tasks while minimizing power consumption (Banbury, Reddi, Lam, Fu, Fazel, Holleman et al., 2020). The big.LITTLE architecture combines high-performance cores with energy-efficient cores, allowing devices to balance performance and power usage based on workload demands (ARM Limited, 2013). NVIDIA's Jetson platform provides energy-efficient GPUs optimized for AI inference at the edge, supporting applications in robotics, drones, and IoT devices (NVIDIA Corporation, 2019).

1.4.2 Software and Algorithmic Optimizations

Optimizing AI models and software frameworks is essential to overcome the computational and memory limitations of edge devices.

Energy-Efficient Computing Platforms

Model compression reduces the size and complexity of AI models, making them more suitable for deployment on edge devices without significant loss in accuracy.

Quantization Quantization involves reducing the precision of the numerical values representing the model's parameters and activations. By converting 32-bit floating-point numbers to lower-bit representations like 8-bit integers, or even 1-bit, models

consume less memory and compute resources (Jacob, Kligys, Chen et al., 2018a). Quantized models not only have smaller storage footprints but also benefit from faster computations on hardware that supports integer arithmetic (Krishnamoorthi, 2018). Techniques such as post-training quantization and quantization-aware training help maintain model accuracy after quantization (Han et al., 2016a).

Pruning Pruning removes redundant or less significant weights and neurons from neural networks, effectively reducing model size and computational requirements (Blalock et al., 2020). Techniques like weight pruning eliminate parameters with minimal impact on the output, while neuron pruning removes entire neurons or filters (Molchanov et al., 2017). Pruning can lead to sparse models that require specialized hardware or algorithms to leverage sparsity for computational gains (Gale et al., 2019). When combined with quantization, pruning can significantly compress models for edge deployment.

Knowledge Distillation Knowledge distillation transfers knowledge from a large, complex model (teacher) to a smaller, more efficient model (student) (Hinton et al., 2015a). The student model is trained to replicate the output of the teacher model, achieving similar performance with fewer parameters. This technique enables the creation of lightweight models suitable for edge devices without substantial accuracy loss. Knowledge distillation is particularly effective for tasks like image classification and natural language processing (Jiao et al., 2020).

Neural Architecture Search Neural architecture search (NAS) plays an important role to find the optimal model architectures for different edge settings.(A. Howard et al., 2019a). MobileVision V3 employs NAS to significantly improve the model performance compared to V2. Once-for-all can search model architectures under different constraints which satisfies the diverse edge device settings.(Cai et al., 2019a).

Matrix factorization Most model weights are from the matrix of Fully-Connected layer. Using the multiplication of two lower-rank to represent the model will reduce

the model size significantly. (Lan, 2019).

Weight clustering it first groups the weights of layers into N clusters, and share the centred value for all the similar weights. This may reduce the weight up to 5x with minimal accuracy reduction (Han, Mao & Dally, 2015).

Frameworks for Edge Deployment

Specialized frameworks facilitate the development and deployment of AI models on edge devices by providing tools optimized for resource constraints.

TensorFlow Lite TensorFlow Lite is a lightweight, cross-platform framework designed for deploying TensorFlow models on mobile and embedded devices [64]. It supports model optimization techniques like quantization and provides hardware acceleration through delegate APIs for GPUs and NPUs. TensorFlow Lite converts models into a specialized FlatBuffer format, reducing size and improving loading times. It also offers an interpreter optimized for on-device inference, enabling developers to run models efficiently on Android, iOS, and embedded Linux platforms [65].

PyTorch Mobile PyTorch Mobile extends the PyTorch framework to mobile and edge environments, allowing developers to run PyTorch models on devices with limited resources [66]. It supports model quantization and provides tools for optimizing and converting models for deployment. PyTorch Mobile integrates with Android and iOS platforms, enabling seamless deployment of AI models within mobile applications. It also supports custom mobile interpreters, reducing application size by including only necessary operators [67].

1.4.3 Co-Design Strategies

Co-design strategies involve jointly optimizing hardware and software components to achieve better performance and efficiency in edge AI systems.

Hardware-Software Co-Optimization

Hardware-software co-optimization focuses on designing algorithms and hardware architectures in tandem to maximize performance and energy efficiency [68]. By understanding the constraints and capabilities of the hardware, software developers can tailor algorithms to leverage specific features, such as parallel processing units or specialized instructions. Conversely, hardware designers can create architectures optimized for the computational patterns of AI algorithms, such as exploiting data reuse and minimizing memory access. This collaborative approach leads to systems where hardware and software are mutually optimized for edge AI applications [69].

Collaborative Development Approaches

Collaborative development approaches involve partnerships between hardware manufacturers, software developers, and AI researchers to create comprehensive solutions for edge AI [70]. Open-source communities and industry consortia play significant roles in advancing edge AI technologies. Initiatives like the MLPerf benchmarking suite provide standardized metrics for evaluating AI performance across different hardware and software configurations [71]. Collaborative efforts help identify bottlenecks, share best practices, and accelerate innovation in edge AI deployments.

1.5. Industrial Adoption and Initiatives in Edge AI

Edge AI has witnessed significant growth due to substantial investments and initiatives by leading technology companies. These companies are developing hardware, software, and platforms to enable efficient AI processing at the edge, thereby transforming industries and driving innovation.

1.5.1 Leading Technology Companies

Several major technology firms are at the forefront of Edge AI development, each contributing unique solutions and strategies.

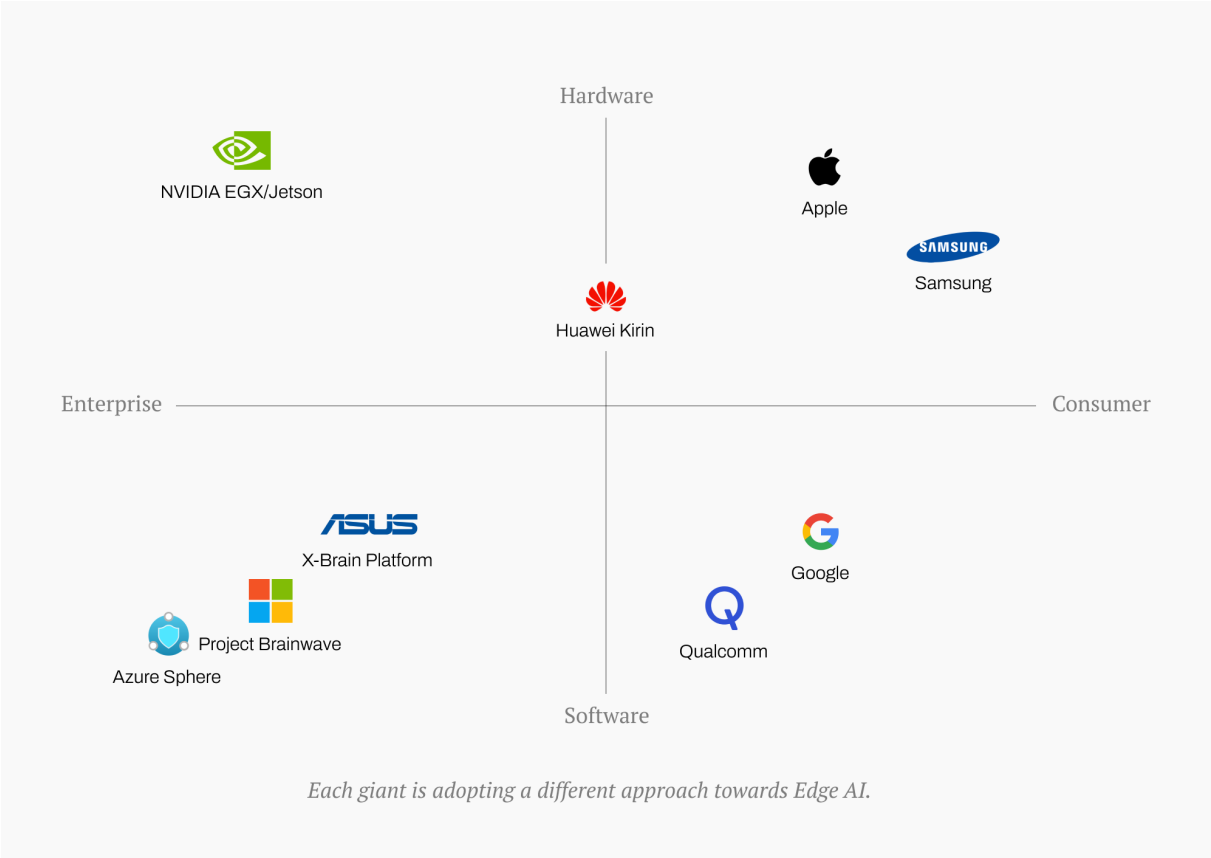

Figure 1.2: Each industry giant has adopted a different approach towards Edge AI

Meta

Formerly known as Facebook, Meta has played a pivotal role in advancing AI technologies, particularly in the open-source domain.

- **Open-Source SLMs and LLMs** Meta has released several open-source Small Language Models (SLMs) and Large Language Models (LLMs) to improve research and development in AI. Notably, in February 2023, Meta introduced the "LLaMA" (Large Language Model Meta AI) collection of foundation language models ranging from 7 billion to 65 billion parameters (Touvron et al., 2023). These models are designed to be efficient and require less computational power, making them more accessible for research and deployment on edge devices.

- **Impact on Edge AI Landscape** The release of LLaMA models has had a significant impact on the Edge AI landscape. By providing high-performing models that are less resource-intensive, Meta has enabled developers to implement advanced AI capabilities on edge devices (Meta AI Research, 2023). This

democratization accelerates innovation in applications like natural language processing, personalized assistants, and real-time translation services on devices with limited computational resources.

Apple

Apple has consistently emphasized privacy and on-device intelligence, integrating AI capabilities directly into its hardware and software ecosystems.

- **On-Device Intelligence and SLMs** Apple's approach centers on performing AI computations locally on devices to enhance privacy and efficiency (Apple Inc., 2021b). The company utilizes SLMs for features such as Siri's speech recognition, QuickType keyboard suggestions, and image processing in the Photos app. The Neural Engine, a dedicated AI processor within Apple's A-series and M-series chips, accelerates machine learning tasks while maintaining energy efficiency (Apple Inc., 2020a). This hardware enables sophisticated AI functions to run seamlessly on devices like iPhones, iPads, and Macs without relying on cloud services.

- **ReaLM LLM Models and Private Compute** Apple has been investing in advanced language models to improve its AI offerings. Although details are scarce due to the company's secretive nature, reports suggest that Apple is developing its own LLMs internally (Gurman, 2023). Additionally, Apple's Private Relay and on-device processing initiatives reflect a commitment to user privacy by minimizing data transmission and processing sensitive information locally (Apple Inc., 2021a).

Google

Google has been a leader in both cloud and edge AI, offering hardware and software solutions that facilitate AI deployment on various platforms.

- **Edge TPU Chips** Google's Edge Tensor Processing Units (Edge TPUs) are specialized ASICs designed to accelerate machine learning tasks on edge devices

(Google Cloud, 2018). These chips enable efficient execution of deep learning models with low latency and power consumption, making them ideal for applications in IoT devices, smart cameras, and embedded systems. Edge TPUs are available through Google's Coral platform, which provides hardware modules and development tools for building edge AI applications (Coral, 2021).

- **ML Kit Platform** ML Kit is Google's mobile SDK that brings machine learning capabilities to Android and iOS apps (Google Developers, 2021). It offers on-device APIs for vision and natural language processing tasks, such as text recognition, face detection, and language translation. By running these models on-device, ML Kit reduces latency and enhances privacy. ML Kit also supports custom model deployment, allowing developers to integrate their own TensorFlow Lite models optimized for mobile devices (TensorFlow, 2021a).

Qualcomm

Qualcomm focuses on enhancing AI processing capabilities in mobile devices through its Snapdragon processors and AI initiatives.

- **Integration of Llama 2 LLMs** In July 2023, Qualcomm announced a collaboration with Meta to optimize and deploy Llama 2-based AI implementations on devices powered by Snapdragon platforms (Qualcomm Technologies, Inc., 2023). This partnership aims to enable on-device generative AI applications, such as intelligent virtual assistants and enhanced productivity tools, without relying on cloud connectivity. By leveraging Qualcomm's AI Engine and Meta's Llama 2 models, the integration seeks to deliver high performance and energy efficiency for AI tasks on smartphones, PCs, and other devices (Qualcomm Technologies, Inc., 2023).

- **Hybrid Edge-Cloud Approach** Qualcomm advocates for a hybrid AI processing model that combines on-device and cloud computing (Qualcomm Technologies, Inc., 2021a). This approach maximizes the benefits of edge processing—like reduced latency and improved privacy—while utilizing cloud resources for more

complex tasks that require greater computational power. The hybrid model allows for scalable AI solutions adaptable to various applications and network conditions, enhancing user experiences across devices (Khemka, 2021).

Other Notable Initiatives

Several other technology companies contribute significantly to Edge AI advancements:

- **Alibaba:** Alibaba Cloud offers edge computing solutions like the Link IoT Edge platform, which integrates AI capabilities for industrial applications (Alibaba Cloud, 2020). The platform enables real-time data processing and analytics at the edge.

- **Samsung:** Samsung incorporates AI accelerators in its Exynos processors, enhancing on-device AI capabilities for tasks such as image recognition and natural language processing (Samsung Electronics, 2021). The company also explores edge AI in smart appliances and IoT devices.

- **Huawei:** Huawei's Ascend series AI processors support edge computing with high-performance capabilities (Huawei, 2021). The Atlas hardware platforms enable AI deployment in areas like smart cities and autonomous driving.

- **Microsoft:** Microsoft's Azure IoT Edge extends cloud intelligence to edge devices, allowing AI models to run locally (Microsoft Azure, 2021a). The platform supports containerized workloads and provides tools for managing edge deployments.

- **NVIDIA:** NVIDIA's Jetson platform offers AI-enabled edge computing devices for various applications (NVIDIA Corporation, 2021a). NVIDIA also provides software frameworks like NVIDIA TensorRT for optimizing AI models for edge deployment.

1.5.2 Impact on the Edge AI Ecosystem

The initiatives by leading technology companies have significantly influenced the Edge AI ecosystem, improving growth, innovation, and collaboration.

Democratization of AI

Open-source models and accessible development tools have democratized AI technologies, allowing a broader range of developers and organizations to implement AI solutions (T. Li et al., 2020). Initiatives like Meta's release of LLaMA models and Google's TensorFlow Lite have lowered barriers to entry, enabling innovation across various sectors. This democratization accelerates the adoption of AI, particularly in industries where resource constraints previously limited technological advancements (Bryant et al., 2008).

Accelerated Innovation and Competition

The investments and technological advancements by major companies have spurred competition in the Edge AI market (K.-F. Lee, 2018). This competition drives rapid innovation, leading to more efficient hardware, optimized algorithms, and novel applications. As companies strive to differentiate their offerings, consumers benefit from improved products and services, such as smarter devices and enhanced user experiences (Y. Chen & Lin, 2021).

Collaboration Between Industry and Academia

Collaborations between industry and academia have been crucial in advancing Edge AI research and development (C. Xu, Liu & Chen, 2021). Partnerships facilitate knowledge exchange, resource sharing, and the development of new technologies. For example, joint research projects focus on areas like model compression techniques, energy-efficient hardware design, and federated learning for privacy-preserving AI (Konečný et al., 2016a). These collaborations contribute to the overall growth and maturity of the Edge AI ecosystem.

	Healthcare	Robotics	Virtual Assistants	Autonomous Driving
Purpose of computation	<150 ms for emergency services (e.g. fall detection)	10-100 ms end-to-end latency, 2 ms data upload latency, and cloud latency often exceeds 100 ms	Should be under 200 ms to avoid degrading user experience	10 ms end-t0-end latency is necessary
Resource Utilization	Sensitive health data	Sensitive household data	Extremely sensitive chat history	Sensitive location data
Incentive Structures	10-50 Mbps for AR/VR applications	Upload rate ranges from 80 Mbps to 12 Gbps depending on neural network architecture	Split AI image recognition may require 144 mbps uplpink	Generates up to 4 TB/ day per vehicle

The requirements for Edge AI applications vary in terms of latency, privacy, and bandwidth.

Figure 1.3: A table summarizing the requirements for Edge AI applications in terms of latency, privacy, and bandwidth.

1.6. Application Domains Driving the Need for Edge AI

Edge AI has become increasingly important due to its ability to address the limitations of cloud-based AI, particularly in applications that require low latency, enhanced privacy, and reduced bandwidth usage. Several domains are driving the demand for Edge AI, including healthcare, robotics, virtual assistants, and autonomous driving. These areas have requirements that necessitate processing data closer to the source.

1.6.1 Healthcare

The healthcare sector stands to benefit significantly from Edge AI, as it enables more immediate and personalized medical services while maintaining patient privacy.

Mobile Health and Personalized Medicine

Mobile health (mHealth) leverages mobile devices and wearable technology to monitor health metrics, deliver medical interventions, and provide personalized healthcare services (S. Wang & Krishnan, 2018a). Personalized medicine tailors treatment to individual patient characteristics, requiring real-time data processing and analysis. Edge AI facilitates mHealth by processing data locally on devices such as smartphones and wearables, enabling continuous monitoring and immediate feedback without the need for constant cloud connectivity (W. Xu et al., 2019a). This is crucial for applications like chronic disease management, where timely interventions can improve patient outcomes.

Edge AI Requirements in Healthcare

Healthcare applications impose strict requirements on AI systems, particularly regarding privacy, latency and bandwidth.

Latency Needs for Critical Responses In medical emergencies, rapid response times are essential. For example, fall detection systems for the elderly must alert caregivers or emergency services immediately after an incident (Chaudhuri et al., 2014). The acceptable latency for such alerts is often less than 150 milliseconds (Patterson et al., 2021a). Edge AI enables real-time processing of sensor data to detect falls and other critical events promptly.

During minimally invasive surgeries, Edge AI can analyze video feeds from laparoscopic or robotic surgical instruments in real time to identify anatomical structures, tissues, and potential areas of concern. For example, the AI can provide visual overlays on a surgeon's display, highlighting blood vessels, nerves, or tumors, helping to avoid accidental damage and ensure more precise cuts. Since the processing is done on edge devices locally in the operating room, this allows for immediate feedback without the latency of cloud-based solutions, which is crucial for time-sensitive surgical procedures. This enhances safety, accuracy, and overall surgical outcomes.

Privacy Regulations and Compliance Healthcare data is highly sensitive and protected by regulations such as the Health Insurance Portability and Accountability Act (HIPAA) in the United States and the General Data Protection Regulation (GDPR) in the European Union (U.S. Department of Health & Human Services, 2013a). Article 9 of GDPR classifies health data as special personal data, requiring explicit consent for its processing (Mach & Becvar, 2017a). Edge AI enhances privacy by keeping patient data on the local device, reducing the risk of data breaches and ensuring compliance with privacy regulations (U.S. Department of Health & Human Services, 2013a). This is particularly important for applications that collect biometric data or personal health information.

Bandwidth Limitations Medical applications involving high-resolution imaging or augmented reality (AR) can consume significant bandwidth (Ding et al., 2019a). For instance, telemedicine consultations using AR/VR technologies may require data rates of 10–50 Mbps (T. Chen et al., 2019). Transmitting this data to the cloud for processing can strain network resources and incur high costs. Edge AI reduces bandwidth usage by processing data locally, transmitting only essential information when necessary (Shi et al., 2016a). This makes advanced medical applications more accessible, even in areas with limited network infrastructure.

Recent Advancements

Several recent developments highlight the growing role of Edge AI in healthcare.

Med-PaLM 2 by Google Google introduced Med-PaLM 2, an LLM fine-tuned on medical datasets to answer medical questions with high accuracy (Singhal et al., 2022). While primarily designed for cloud deployment, efforts are underway to optimize such models for edge devices, enabling doctors and patients to access advanced diagnostic tools offline or in low-connectivity environments.

Collaborative Efforts (Fitbit and Google Research) Fitbit, in collaboration with Google Research, is developing an LLM specifically for personalized health and well-

ness (Fitbit News, 2022). The goal is to provide users with summaries and recommendations based on data collected from their wearable devices. By processing this data locally, users benefit from personalized insights without compromising their privacy.

On-Device Health Monitoring (Apple Watch) Apple has integrated health-focused AI capabilities into the Apple Watch, utilizing on-device machine learning for features like Fall Detection, Irregular Rhythm Notification, and Blood Oxygen monitoring (International Energy Agency, 2021). These features operate independently of cloud services, ensuring immediate responses and safeguarding user data.

1.6.2 Robotics

Robotics is another domain where Edge AI is essential, particularly as robots become more integrated into daily life and industrial processes.

Integration of LLMs in Robotics

The advent of LLMs like GPT-3 and GPT-4 has spurred interest in integrating advanced language understanding into robotic systems (Liang & Lee, 2022). This integration enables robots to interpret complex instructions, engage in natural language interactions, and perform tasks that require contextual understanding.

Edge AI Requirements in Robotics

Robotic applications impose stringent requirements on AI systems, necessitating real-time processing, data privacy, and efficient bandwidth usage.

Real-Time Processing and Latency Constraints Robots operating in dynamic environments must process sensor data and make decisions in real-time (Ahmad & Lee, 2020). Latency in control loops can lead to suboptimal performance or safety hazards. The 3rd Generation Partnership Project (3GPP) standards indicate that 5G remote-controlled robotics require end-to-end latency between 10–100 milliseconds and an intermediate data uploading latency of 2 milliseconds (B. Li, Li & Liu, 2018a). Edge

AI allows robots to process data locally, reducing latency and improving responsiveness (Wan et al., 2018a).

Data Privacy Domestic robots, such as home assistants or cleaning robots, often operate in personal spaces and may collect sensitive data (Siau & Wang, 2018). Processing this data on-device protects user privacy by preventing transmission of personal information over networks. Edge AI ensures that data such as images, voice recordings, or behavioral patterns remain confidential, addressing privacy concerns and building user trust (Tung, 2018).

Bandwidth Usage Robots equipped with multimodal sensors (e.g., cameras, LiDAR, tactile sensors) generate large volumes of data (J. Zhang & Tao, 2020) Transmitting this data to the cloud for processing is bandwidth-intensive and may not be feasible in environments with limited connectivity. 3GPP standards suggest that for split inference, the necessary upload data rate ranges from 80 Mbps to 12 Gbps, depending on the neural network architecture (European Parliament and Council of European Union, 2016b). Edge AI mitigates bandwidth issues by processing sensor data locally, transmitting only essential information when needed (Mahmoud & Mohamad, 2019).

Case Studies and Implementations

- **NVIDIA Isaac Platform:** NVIDIA's Isaac platform provides a suite of tools and hardware for developing AI-powered robots with edge processing capabilities (NVIDIA Corporation, 2021b). It enables real-time perception, navigation, and manipulation tasks.
- **Boston Dynamics:** Robots like Spot and Atlas use onboard processing for real-time control and obstacle avoidance (Boston Dynamics, 2021). Edge AI allows these robots to operate autonomously without reliance on cloud connectivity.
- **Amazon Astro:** Amazon's home robot, Astro, utilizes edge computing to perform tasks like home monitoring and personal assistance while preserving user privacy (Amazon, 2021).

1.6.3 Virtual Assistants

Virtual assistants have become ubiquitous, providing users with voice-activated assistance and personalized services.

Evolution Post-ChatGPT

The release of ChatGPT demonstrated the potential of LLMs in generating human-like text and engaging in complex conversations (OpenAI, 2022). This sparked a surge in the development of virtual assistants capable of more natural and context-aware interactions. Companies are exploring ways to deploy LLMs on edge devices to enhance virtual assistants while addressing privacy and performance concerns (Z. Chen et al., 2022a).

Edge AI Requirements in Virtual Assistants

Key requirements for virtual assistants include low latency, privacy protection, and efficient bandwidth usage.

User Experience and Latency Virtual assistants need to respond promptly to user queries to provide a seamless experience (X. Lu & Li, 2020a). NVIDIA research indicates that end-to-end latency exceeding 200 milliseconds becomes noticeable to users, potentially degrading interaction quality (S. Wang et al., 2017). Edge AI reduces latency by processing voice recognition and natural language understanding locally, enabling quicker responses and improving user satisfaction (He et al., 2019a).

Handling Sensitive User Data Privately Virtual assistants often process personal information, including contacts, messages, and search history (Hoy, 2018a). Processing this data on-device enhances privacy by minimizing data exposure and reducing the risk of unauthorized access. Edge AI ensures that sensitive information remains on the user's device, aligning with privacy regulations and increasing user trust (Shearer & Gottfried, 2017).

Reducing Bandwidth for High User Volume With millions of users interacting with virtual assistants, transmitting data to cloud servers can lead to network congestion and high operational costs (Ericsson, 2020). According to 3GPP, split AI image recognition may require an uplink data rate of 144 Mbps (J. Lin et al., 2017a). Edge AI alleviates bandwidth issues by handling data processing locally, making virtual assistant services more scalable and cost-effective (X. Li & Wang, 2020).

Edge-Based Virtual Assistant Models

- **Apple Siri:** Apple's Siri leverages on-device processing for voice recognition and natural language tasks, improving responsiveness and privacy (Apple Inc., 2020d).
- **Google Assistant:** Google has introduced on-device speech recognition models to enhance the performance of Google Assistant, reducing reliance on cloud processing (Schuster et al., 2020a).
- **Amazon Alexa Voice Service (AVS) Integration for Edge Devices:** Amazon provides tools for integrating Alexa into devices with edge capabilities, allowing for local processing of voice commands (Amazon Developer Services, 2021).

1.6.4 Autonomous Driving

Autonomous vehicles (AVs) represent a complex domain requiring real-time processing and advanced AI capabilities.

Limitations of Modular Architectures

Traditional AV systems use modular architectures, dividing tasks into perception, prediction, and planning modules (T. Chen et al., 2020a). While this approach simplifies development, it limits the system's ability to perform holistic reasoning and handle complex, unstructured scenarios. LLMs offer the potential to enhance AV systems by integrating knowledge across modules, enabling more sophisticated decision-making

(B. Wu et al., 2021). However, deploying LLMs in vehicles presents challenges due to computational constraints.

Edge AI Requirements in Autonomous Vehicles

AVs require AI systems that meet strict latency, privacy, and data management requirements.

Safety-Critical Latency Requirements AVs must process sensor data and make driving decisions within milliseconds to ensure safety (Kugler, 2018). 3GPP specifies that autonomous driving scenarios may necessitate end-to-end latency of 10 milliseconds (Premsankar et al., 2018b). Edge AI enables real-time processing of sensor inputs, such as camera feeds, LiDAR, and radar data, without the delays associated with cloud communication (Y. Liu et al., 2020).

Privacy Requirements Transmitting vehicle data to cloud servers can expose sensitive information, such as location history and driving patterns (Arvin et al., 2014a). This raises privacy concerns for users and potential regulatory issues. By processing data on-board, Edge AI protects user privacy and complies with data protection regulations (European Data Protection Board, 2019).

Managing Massive Data Generation AVs generate vast amounts of data, up to 4 terabytes per day per vehicle (Sood & Mahajan, 2017). Uploading this data to the cloud is impractical due to bandwidth limitations and costs. Edge AI allows AVs to process and analyze data locally, transmitting only essential information when necessary (X. Ma et al., 2018a). This reduces the burden on network infrastructure and improves system efficiency.

Incorporating LLMs into Vehicle Systems

Integrating LLMs into AVs can enhance capabilities such as:

- **Natural Language Interaction:** Enabling passengers to communicate with

the vehicle using natural language for navigation, entertainment, or control functions (Ge et al., 2021).

- **Contextual Understanding:** Improving the vehicle's ability to interpret complex driving scenarios and make more informed decisions (Kapania & Gerdes, 2015).

- **Personalization:** Tailoring driving experiences based on user preferences and behaviors (Reddy & Chandra, 2020). Companies like Tesla are exploring the use of advanced AI models within vehicles to enhance autonomy and user experience (Tesla, 2021). Edge AI is critical for deploying these models within the computational constraints of vehicle hardware.

1.7. Cross-Domain Requirements for Edge AI Applications

Edge AI applications span various domains, each with unique requirements. However, certain fundamental needs are consistent across these domains. Latency, privacy and security, and bandwidth constraints are critical considerations that influence the design and deployment of Edge AI solutions. Understanding and addressing these cross-domain requirements are essential for the effective implementation of Edge AI technologies.

1.7.1 Latency Considerations

Tolerable Latency Thresholds Across Applications

Different applications have varying tolerable latency thresholds depending on their specific requirements and the consequences of delays.

- **Healthcare Applications:** In critical healthcare scenarios like remote surgery or emergency response systems, latency must be minimized to prevent adverse

outcomes. Latency thresholds are often less than 100 milliseconds to ensure real-time feedback and control (Cisco Systems, 2018).

- **Autonomous Vehicles:** Self-driving cars require ultra-low latency to process sensor data and make driving decisions instantaneously. Latency must be within 10 milliseconds to respond to dynamic driving conditions and avoid collisions (J. Zhang & Chen, 2020a).
- **Industrial Automation:** Manufacturing processes using robotics and automation systems need latencies below 1 millisecond to synchronize operations and maintain precision (Y. Lu et al., 2020).
- **Virtual Reality (VR) and Augmented Reality (AR):** For immersive experiences, latency should be less than 20 milliseconds to prevent motion sickness and ensure a seamless user experience (S. Zhan & Kojima, 2017).
- **Financial Trading:** High-frequency trading systems demand latencies in the microseconds range to capitalize on market fluctuations (Aldridge, 2013a).
- **Consumer Applications:** Virtual assistants and mobile applications aim for latencies below 200 milliseconds to maintain a responsive and engaging user experience (Bixby & Renaudin, 2019a).

Understanding these thresholds helps in designing Edge AI systems that meet the specific latency requirements of each application domain.

Techniques for Latency Reduction

To achieve the necessary latency levels, various techniques are employed in Edge AI systems:

- **On-Device Processing:** Performing computations locally on the device eliminates the need for data transmission to remote servers, significantly reducing latency (Satyanarayanan, 2017b).

- **Efficient Model Design:** Developing lightweight and optimized AI models that require fewer computational resources speeds up processing times (A. G. Howard et al., 2017b).
- **Hardware Acceleration:** Utilizing specialized hardware such as GPUs, TPUs, NPUs, and FPGAs accelerates AI computations, decreasing inference time (Y.-H. Chen et al., 2017a). Edge Caching: Storing frequently accessed data or preprocessed information at the edge reduces retrieval times (X. Li et al., 2018).
- **Network Optimization:** Implementing efficient communication protocols and network configurations minimizes transmission delays (B. Cheng et al., 2018a).
- **Parallel Processing:** Leveraging multi-core processors and parallel computing techniques enhances processing speed (Y. Kang, Hauswald, Gao et al., 2017).
- **Compression and Quantization:** Reducing the size of AI models through compression and quantization techniques decreases processing time and memory usage (Y. Choi et al., 2018).

By integrating these techniques, Edge AI applications can meet the stringent latency requirements across various domains.

1.7.2 Privacy and Security

Privacy and security are paramount in Edge AI applications, especially when handling sensitive data. Ensuring data protection and compliance with regulations is essential for maintaining user trust and preventing unauthorized access.

Data Protection Regulations

Several regulations govern data protection, impacting how Edge AI systems are designed and operated:

- **General Data Protection Regulation (GDPR):** Enforced in the European Union, GDPR mandates strict guidelines on personal data processing, requiring

explicit user consent and data minimization (European Union, 2016). Edge AI facilitates compliance by processing data locally, reducing the need to transfer personal data to centralized servers.

- **Health Insurance Portability and Accountability Act (HIPAA):** In the United States, HIPAA sets standards for protecting sensitive patient health information (U.S. Department of Health & Human Services, 2013b). Edge AI in healthcare applications can enhance compliance by keeping patient data on-device.

- **California Consumer Privacy Act (CCPA):** CCPA grants California residents rights over their personal information and imposes obligations on businesses handling such data (California Legislative Information, 2018). Edge AI can help businesses comply by minimizing data collection and processing data locally.

- **Children's Online Privacy Protection Act (COPPA):** COPPA regulates the online collection of personal information from children under 13 (Federal Trade Commission, 1998). Edge AI applications targeting minors must ensure data protection and compliance with these regulations.

Secure On-Device Processing

Ensuring security in on-device processing involves several strategies:

- **Encryption:** Implementing robust encryption protocols for data at rest and in transit protects against unauthorized access (Alrawais et al., 2017a).

- **Secure Boot and Trusted Execution Environments (TEEs):** Devices can use secure boot processes and TEEs to verify software integrity and isolate sensitive computations (Sabt et al., 2015).

- **Anonymization and Data Minimization:** Processing only the necessary data and anonymizing personal identifiers reduce the risk of exposing sensitive information (Gai et al., 2017).

- **Regular Security Updates:** Keeping device firmware and software updated patches vulnerabilities and enhances security (Y. Liu et al., 2014).
- **Biometric Authentication:** Using biometric methods like fingerprint or facial recognition adds an extra layer of security for accessing devices and applications (Jain et al., 2011).
- **Edge AI Frameworks with Security Features:** Utilizing AI frameworks that offer built-in security features, such as secure model deployment and encrypted model files (Rajendran et al., 2012).

By implementing these measures, Edge AI applications can provide secure and private services.

1.7.3 Bandwidth and Network Constraints

Bandwidth limitations and network constraints significantly impact the performance and scalability of Edge AI applications. Efficient bandwidth usage is essential to reduce costs and ensure consistent service quality.

Impact on Network Infrastructure

Edge AI can alleviate pressure on network infrastructure in several ways:

- **Reduced Data Transmission:** By processing data locally, Edge AI minimizes the need to transmit large volumes of raw data over networks (M. Chen et al., 2014). This reduces network congestion and lowers the demand on bandwidth.
- **Scalability:** Offloading processing tasks to edge devices allows networks to support more devices and users without significant infrastructure upgrades (Mach & Becvar, 2017b).
- **Latency Reduction:** Decreasing reliance on centralized servers reduces round-trip times, improving overall network performance (Shi et al., 2016b).

- **Localized Content Delivery:** Edge caching and content delivery networks (CDNs) store content closer to users, enhancing access speeds and reducing backbone network load (Y. Li & Liu, 2018a).

However, as the number of edge devices increases, there is a need for efficient network management and coordination to prevent interference and maintain service quality (Omoniwa et al., 2018a).

Cost Implications for Users and Providers

Efficient bandwidth usage has direct cost benefits:

- **Cost Savings for Users:** Users benefit from reduced data usage and lower network charges, especially in regions where data costs are high or data caps are enforced (C. Wang et al., 2018).
- **Operational Efficiency for Providers:** Service providers can reduce operational costs by decreasing the amount of data transmitted and processed in centralized data centers (Taleb et al., 2017a).
- **Infrastructure Investment:** By leveraging Edge AI, providers may delay or reduce the need for expensive infrastructure upgrades to handle increased data traffic (Y. Mao, Zhang & Letaief, 2017).
- **Energy Consumption:** Lower data transmission reduces energy consumption in network equipment, contributing to cost savings and environmental benefits (Peng et al., 2018).
- **Revenue Opportunities:** Providers can offer new services and monetize edge computing capabilities, creating additional revenue streams (K. Zhang et al., 2016b).

1.8. Challenges and Future Directions

1.8.1 Technical Challenges

Computational Limitations of Edge Devices

Edge devices, such as smartphones, IoT sensors, and embedded systems, often have limited computational resources compared to cloud servers. These constraints include lower processing power, memory, and storage capacity, which make it challenging to run complex AI models (Y. Li et al., 2020). Optimizing models to fit within these limitations without significant loss of accuracy is a critical challenge.

Energy Efficiency and Battery Life

Many edge devices are battery-powered and require energy-efficient operation to prolong battery life. Running AI computations can be power-intensive, leading to rapid battery depletion (C. Xu, Liu, Li et al., 2021). Developing energy-efficient algorithms and hardware is essential to ensure that Edge AI applications are sustainable and practical for everyday use.

Model Accuracy vs. Resource Consumption

Balancing model accuracy with resource consumption is a significant challenge. Highly accurate AI models tend to be larger and require more computational resources, which may not be feasible on edge devices (J. Choi et al., 2018). Techniques such as model compression, quantization, and pruning can reduce model size but may also impact performance. Achieving an optimal trade-off is an ongoing area of research.

1.8.2 Ethical and Regulatory Considerations

As Edge AI becomes more pervasive, ethical issues and regulatory compliance become increasingly important.

Ensuring Data Privacy and User Consent

While Edge AI enhances privacy by processing data locally, it still requires careful handling of personal data (European Union, 2016). Developers must implement robust security measures to prevent unauthorized access and comply with data protection regulations such as GDPR and HIPAA. Transparent data handling practices and user education are also important to maintain trust.

Addressing Bias and Fairness in Edge AI Models

AI models can inadvertently perpetuate biases present in the training data (Mehrabi et al., 2021). Deploying biased models on edge devices can lead to unfair or discriminatory outcomes. Ensuring fairness and addressing bias in Edge AI models is an ethical imperative that requires careful consideration during model development and deployment. This includes diversifying training datasets and implementing bias mitigation techniques.

1.8.3 Research Opportunities

Advancements in Model Compression

Research into advanced model compression techniques, such as quantization, pruning, and knowledge distillation, can help create efficient models suitable for edge devices without significant loss of accuracy (Z. Wang et al., 2020). Continued innovation in this area will enable more complex AI applications to run on resource-constrained devices, expanding the capabilities of Edge AI.

Edge-to-Cloud Collaborative AI

Developing frameworks for seamless collaboration between edge devices and cloud servers can optimize performance and resource utilization (J. Liu et al., 2020). Edge-to-cloud synergy allows for complex tasks to be distributed appropriately, enhancing the capabilities of Edge AI applications while leveraging the strengths of both edge and cloud computing.

Standardization and Interoperability

Establishing standards and protocols for Edge AI can facilitate interoperability between devices and platforms (Murshed et al., 2019). Standardization efforts can accelerate adoption, improve security, and create a more cohesive Edge AI ecosystem.

1.9. Conclusion

1.9.1 Recap of the Necessity for Edge AI

The rapid advancement of artificial intelligence has ushered in powerful models and applications that demand significant computational resources and real-time processing capabilities. Traditional cloud-based AI deployments face challenges such as high latency, privacy concerns, and bandwidth limitations, which hinder their effectiveness in time-sensitive and privacy-critical applications (Satyanarayanan, 2017c). As discussed in this chapter, Edge AI addresses these challenges by processing data locally on edge devices, thereby reducing latency, enhancing data privacy, and lowering bandwidth usage (Shi et al., 2016c). The necessity for Edge AI is evident across various domains. In healthcare, real-time monitoring and immediate responses are crucial for patient outcomes (S. Wang & Krishnan, 2018a). Robotics requires instantaneous processing for dynamic environments (Wan et al., 2018a), while virtual assistants benefit from reduced latency and enhanced privacy (He et al., 2019a). Autonomous driving demands ultra-low latency and real-time decision-making to ensure safety (J. Zhang & Chen, 2020a). These applications underscore the importance of Edge AI in meeting the stringent requirements that cloud-based solutions cannot adequately address.

1.9.2 The Transformative Potential of Edge AI

Edge AI holds transformative potential by enabling intelligent applications that are responsive, private, and efficient. By bringing computation closer to the data source, Edge AI empowers new use cases and enhances existing ones. In healthcare, it facilitates personalized medicine and timely interventions while maintaining patient con-

fidentiality (N. Rieke et al., 2020). In robotics and autonomous systems, Edge AI enhances autonomy and responsiveness, enabling robots and vehicles to operate safely and efficiently without constant cloud connectivity (NVIDIA Corporation, 2021b).

Advancements in hardware innovations, such as specialized edge AI chips and integrated NPUs, coupled with software optimizations like model compression techniques, have made it feasible to deploy sophisticated AI models on resource-constrained devices (Sze, Chen et al., 2017b). Collaborative efforts among leading technology companies are driving innovation and democratizing access to AI technologies, further accelerating the adoption of Edge AI (T. Li et al., 2020).

1.9.3 Final Thoughts on the Future Landscape

Looking ahead, Edge AI is set to play a pivotal role in the future of artificial intelligence. Ongoing research and development efforts aimed at overcoming technical challenges—such as computational limitations and energy efficiency—will further enhance the capabilities of edge devices (Z. Wang et al., 2020). Addressing ethical considerations, including data privacy, security, and fairness, is essential to build trust and ensure responsible deployment of Edge AI technologies (Mehrabi et al., 2021).

The convergence of Edge AI with other emerging technologies, such as 5G networks and the Internet of Things (IoT), will create new opportunities and applications (N. Zhang et al., 2018). As Edge AI continues to evolve, it will enable smarter cities, more efficient industries, and improved quality of life. By harnessing the transformative potential of Edge AI, we can pave the way for a future where intelligent systems are seamlessly integrated into every aspect of society, delivering benefits that are immediate, personalized, and secure.

Chapter 2

What is Edge AI

2.1. Introduction

2.1.1 Purpose and Scope of the Chapter

Edge Artificial Intelligence (Edge AI) represents a transformative shift in how data is processed and analyzed, bringing computational intelligence directly to local devices rather than relying solely on centralized cloud infrastructures. This chapter aims to provide a comprehensive understanding of Edge AI by exploring its definitions, historical evolution, key characteristics, and significance in the modern technological landscape.

Objectives The primary objectives of this chapter are:

- **To define Edge AI** and distinguish it from other computing paradigms such as Cloud AI and Distributed AI.
- **To trace the historical development of Edge AI,** identifying the technological advancements that have enabled its emergence.
- **To discuss the drivers behind the adoption of Edge AI** and its importance in contemporary technology and industry.

2.1.2 The Emergence of Edge AI

Historical Context and Evolution The concept of processing data at the edge of the network is not entirely new; it has roots in earlier computing paradigms such as distributed computing and mobile computing (Shi et al., 2016e). However, the term "Edge AI" has gained prominence with the rapid advancements in AI algorithms, increased computational power of edge devices, and the growing need for real-time data processing. In the early days of computing, most computational tasks were handled by centralized mainframes and servers. With the advent of cloud computing, organizations began leveraging scalable resources to process vast amounts of data remotely (Armbrust et al., 2010b). Cloud computing offered significant advantages in terms of scalability and resource management but also introduced challenges related to latency, bandwidth, and privacy (Agency, 2021a; Shi et al., 2016e). As the number of connected devices grew exponentially, transmitting all data to the cloud became impractical due to:

- **Latency constraints:** Critical applications could not tolerate the delays introduced by data transmission to and from remote servers (J. Chen & Ran, 2019; Shi et al., 2016e).
- **Bandwidth limitations:** The sheer volume of data generated by Internet of Things (IoT) devices strained network capacities (Cisco, 2020c; Systems, 2018a).
- **Privacy and security concerns:** Sending sensitive data over networks increased the risk of breaches and raised compliance issues with data protection regulations (Parliament & of European Union, 2016; Union, 2016a).

Edge computing emerged as a solution by bringing computation closer to the data source, reducing the need for data transmission to the cloud (Satyanarayanan, 2017d; Shi et al., 2016e). Edge AI builds upon this concept by integrating artificial intelligence capabilities into edge devices, enabling them to process data and make intelligent decisions locally (J. Chen & Ran, 2019; Satyanarayanan, 2017d). Key milestones in the evolution of Edge AI include:

- **Advancements in hardware:** The development of more powerful and energy-efficient processors, such as NVIDIA's Jetson series (Corporation, 2021) and Google's Edge TPU (Cloud, 2018), has enabled complex AI models to run on edge devices.

- **Optimization of AI models:** Techniques like model compression, quantization, and pruning have made it possible to deploy AI models on devices with limited resources (L. Deng et al., 2020b; Han et al., 2016c; Jacob, Kligys, Chen et al., 2018b).

- **Emergence of lightweight frameworks:** AI frameworks such as TensorFlow Lite (TensorFlow, 2021b) and PyTorch Mobile (PyTorch, 2020) have facilitated the deployment of models on mobile and embedded devices. These developments have collectively enabled the practical implementation of Edge AI, transforming it from a theoretical concept into a viable technology impacting various sectors.

Drivers Behind Edge AI Adoption Several factors have accelerated the adoption of Edge AI:

- **Need for real-time processing:** Applications like autonomous vehicles (B. Li, Li & Liu, 2018b), industrial automation (B. Cheng et al., 2018c), and healthcare monitoring (S. Wang & Krishnan, 2018c) require immediate responses that cannot tolerate the latency introduced by cloud processing.

- **Privacy and security concerns:** Processing data locally reduces the risk of data breaches and helps comply with data protection regulations like GDPR (Parliament & of European Union, 2016; Union, 2016a) and HIPAA (of Health & Human Services, 2013a), enhancing user privacy and security (Alrawais et al., 2017b).

- **Bandwidth limitations:** Transmitting large volumes of data to the cloud is costly and impractical, especially in areas with limited connectivity. Edge AI

reduces reliance on constant network access by processing data locally (Cisco, 2020c; Systems, 2018a).

- **Scalability:** As the number of IoT devices increases, cloud infrastructures may struggle to handle the data load. Edge AI distributes the processing burden, alleviating pressure on centralized servers (Shi et al., 2016e).
- **Technological advancements:** Improvements in hardware and software have made it feasible to deploy AI models on edge devices, making Edge AI a practical solution for various applications (Banbury, Reddi, Lam et al., 2020; L. Deng et al., 2020b).

2.1.3 Importance of Edge AI in Modern Technology

Edge AI is revolutionizing the way data is processed and analyzed, offering significant advantages over traditional cloud-based approaches:

- **Enhanced performance:** By processing data locally, Edge AI reduces latency, leading to faster decision-making and improved user experiences in applications like virtual assistants (Hoy, 2018b) and real-time translation services (X. Lu & Li, 2020b).
- **Improved privacy and security:** Local data processing minimizes the exposure of sensitive information, addressing privacy concerns and helping organizations comply with regulations (Parliament & of European Union, 2016; Union, 2016a).
- **Cost efficiency:** Reducing data transmission to the cloud lowers bandwidth costs and decreases the need for expensive infrastructure investments (Cisco, 2020c; Systems, 2018a).
- **Enabling new applications:** Edge AI makes it possible to deploy AI in environments where cloud connectivity is limited or unreliable, expanding the reach of AI technologies to remote and mobile settings (Ding et al., 2019b; Shi et al., 2016e).

- **Sustainability:** By decreasing reliance on large data centers, Edge AI can contribute to reducing the environmental impact associated with high energy consumption in cloud computing (Agency, 2021a; Patterson et al., 2021b).

Edge AI is becoming increasingly critical in industries such as:

- **Healthcare:** Enabling real-time patient monitoring and personalized medicine while ensuring data privacy (S. Wang & Krishnan, 2018c).
- **Automotive:** Supporting autonomous driving features and enhancing vehicle safety systems (B. Li, Li & Liu, 2018b).
- **Manufacturing:** Facilitating predictive maintenance and optimizing production processes through real-time analytics (B. Cheng et al., 2018c).
- **Consumer electronics:** Enhancing user experiences in smartphones, wearables, and smart home devices through on-device intelligence (Inc., 2020a, 2020c).

The integration of AI at the edge empowers devices to be more responsive, intelligent, and capable of operating independently, driving innovation and creating new opportunities across various sectors.

2.2. Fundamentals of Edge AI

2.2.1 Definition and Concept of Edge AI

Edge Artificial Intelligence (Edge AI) refers to the deployment of AI algorithms and models directly on devices at the edge of the network, such as sensors, smartphones, and embedded systems, enabling data processing and decision-making close to the data source (Shi et al., 2016e). Unlike traditional AI systems that rely heavily on centralized cloud computing, Edge AI brings computation and intelligence to the edge devices themselves. This paradigm shift allows for real-time analytics, reduced latency, enhanced privacy, and improved efficiency in data handling(Satyanarayanan, 2017d; Shi et al., 2016e).

Edge AI integrates the capabilities of edge computing and artificial intelligence to process data locally, without the need for constant communication with centralized servers.(J. Chen & Ran, 2019; Shi et al., 2016e). By leveraging the computational power of modern edge devices and optimized AI models, Edge AI systems can perform complex tasks such as image recognition, natural language processing, and predictive analytics directly on the device (Banbury, Reddi, Lam et al., 2020; L. Deng et al., 2020b).

2.2.2 Key Characteristics of Edge AI

Processing at the Edge Devices Edge AI's foundational characteristic is local data processing. Data generated by edge devices is processed on-site, reducing the need to transmit large volumes of raw data to centralized servers (J. Chen & Ran, 2019; Shi et al., 2016e). This local processing minimizes dependency on network connectivity and alleviates network congestion (Cisco, 2020a; Systems, 2018b). For instance, a smart camera equipped with Edge AI can analyze video feeds in real-time to detect anomalies without sending the entire video stream to the cloud (J. Chen & Ran, 2019; B. Li, Li & Liu, 2018b).

Real-Time Decision Making By processing data locally, Edge AI enables real-time decision-making capabilities (J. Chen & Ran, 2019; S. Deng et al., 2020c). This is critical in applications where immediate responses are essential, such as autonomous vehicles that must react to road conditions instantaneously (B. Li, Li & Liu, 2018b), industrial control systems that require precise timing (B. Cheng et al., 2018b), and healthcare devices monitoring patient vitals (S. Wang & Krishnan, 2018c). The elimination of network latency ensures that decisions are made promptly, enhancing system responsiveness and reliability (J. Chen & Ran, 2019; Satyanarayanan, 2017d).

Enhanced Privacy and Security Edge AI enhances privacy and security by keeping sensitive data on the device rather than transmitting it over networks (Alrawais et al., 2017b; Omoniwa et al., 2018b). Local data processing reduces exposure to potential

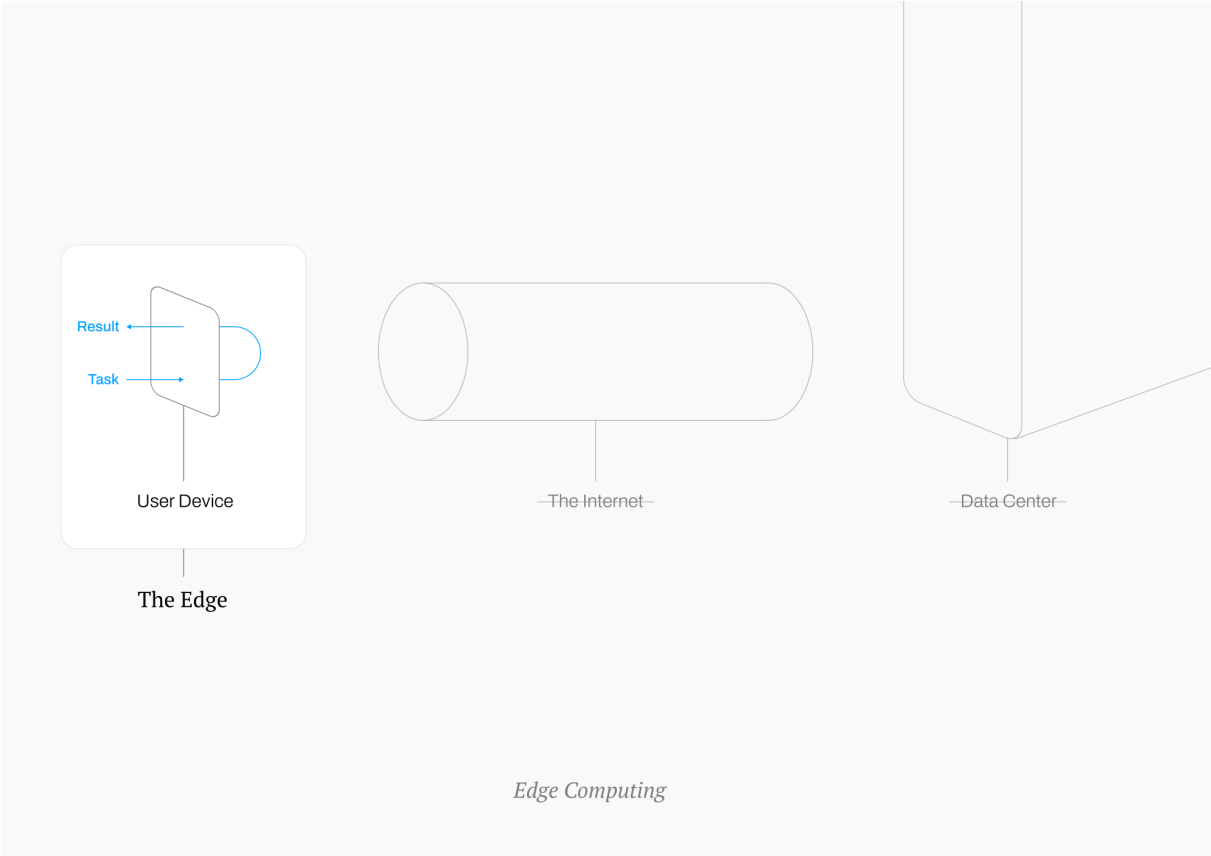

Figure 2.1: The figure showcases computations on the edge

cyber threats during data transmission and complies with data protection regulations like GDPR (Union, 2016b) and HIPAA (of Health & Human Services, 2013b). Applications in healthcare (X. Ma et al., 2018b; S. Wang & Krishnan, 2018c), finance (Aldridge, 2013b), and personal devices (Abomhara & Køien, 2015; Inc., 2020c) benefit significantly from the increased privacy provided by Edge AI.

Reduced Latency and Bandwidth Usage Edge AI reduces latency by eliminating the need to send data to remote servers for processing (Shi et al., 2016e, 2016f). This is especially important for time-sensitive applications where delays can lead to suboptimal outcomes or safety risks (Bixby & Renaudin, 2019b; B. Li, Li & Liu, 2018b). Additionally, by processing data locally, Edge AI minimizes bandwidth usage, reducing costs associated with data transmission and mitigating network bottlenecks (Cisco, 2020a; Systems, 2018b). This efficiency is crucial in environments with limited or expensive connectivity options (Shi & Dustdar, 2016; Shi et al., 2016e).

Autonomy and Decentralization Edge AI promotes autonomy by enabling devices to operate independently of centralized infrastructure (S. Deng et al., 2020c; Shi et al., 2016e). Decentralization allows edge devices to function even in the absence of network connectivity, making them suitable for remote or mobile applications like drones, remote sensors, and autonomous vehicles (Arvin et al., 2014b; B. Li, Li & Liu, 2018b). This autonomy enhances system robustness and scalability by distributing computational workloads across multiple devices (S. Deng et al., 2020c; Satyanarayanan, 2017d).

2.2.3 Components of Edge AI Systems

An Edge AI system comprises several components that work together to enable local data processing and intelligent decision-making.

Edge Devices Edge devices are the physical hardware where AI models are deployed. They vary widely in capabilities and include:

- **Sensors and IoT Devices:** Sensors and IoT devices are often resource-constrained but play a critical role in data collection (J. Lin et al., 2017b; Omoniwa et al., 2018b). Examples include environmental sensors, smart thermostats, and industrial monitoring equipment. Advances in microcontrollers and embedded processors have enabled these devices to perform basic AI tasks, such as anomaly detection and pattern recognition (Banbury, Reddi, Lam, Fu, Fazel, Holleman & Whatmough, 2020; Ding et al., 2019c). For instance, a vibration sensor on industrial machinery can locally analyze data to predict maintenance needs (J. Kang et al., 2017; Wan et al., 2018b).

- **Smartphones and Wearables:** Modern smartphones and wearable devices are equipped with powerful processors, GPUs, and dedicated AI hardware like Apple's Neural Engine (Apple Inc., 2020b; Inc., 2020b) and Qualcomm's Snapdragon AI Engine (Qualcomm Technologies, Inc., 2021b). These devices support complex AI applications such as facial recognition (Inc., 2020b), voice assistants

(Inc., 2020c), and health monitoring (S. Wang & Krishnan, 2018c). On-device AI capabilities enhance user experiences by providing faster responses and improved privacy (Z. Chen et al., 2022b; Inc., 2020c).

- **Embedded Systems:** Embedded systems are specialized computing systems that perform dedicated functions within larger systems (Y.-H. Chen et al., 2017c; Y. Li & Liu, 2018c). They are integral to applications like automotive systems, robotics, and smart appliances. Edge AI enables embedded systems to process data locally for tasks such as autonomous navigation (T. Chen et al., 2020d; B. Li, Li & Liu, 2018b), robotic control (Y.-H. Chen et al., 2017c; NVIDIA Corporation, 2021c), and smart home automation (B. Cheng et al., 2018b; Y. Li & Liu, 2018c).

Edge Computing Infrastructure Edge computing infrastructure provides the foundational support for Edge AI deployments, including hardware accelerators, networking components, and middleware (Shi et al., 2016e, 2016f). Components include:

- **Edge Servers and Gateways:** Devices that aggregate data from multiple edge devices, provide additional processing power, and facilitate communication between the edge and cloud (Y. Li & Liu, 2018b; Taleb et al., 2017b).
- **Networking Infrastructure:** Technologies that enable efficient communication within the edge network and between edge and cloud, such as 5G, Wi-Fi, and specialized protocols (Taleb et al., 2017b; J. Zhang & Chen, 2020b).
- **Middleware and Software Platforms:** Software layers that provide services like device management, security, data analytics, and application deployment (Microsoft Azure, 2021b; Yi et al., 2015).

2.2.4 Edge AI Workflow

The Edge AI workflow encompasses the processes from data generation to action execution, enabling intelligent decision-making at the edge.

Data Generation and Collection Edge devices generate and collect data from their environment using sensors and input mechanisms (J. Lin et al., 2017b; Omoniwa et al., 2018b). This data can include images, audio, environmental readings, user interactions, and more. For example, a wearable health device collects biometric data like heart rate and activity levels (S. Wang & Krishnan, 2018c; W. Xu et al., 2019b).

Local Processing and Analysis Collected data is processed and analyzed locally using AI models deployed on the edge device (J. Chen & Ran, 2019; Shi et al., 2016e). Processing steps may include:

- **Preprocessing:** Cleaning and formatting data for analysis (Bishop, 2006).
- **Inference:** Running AI models to extract insights from data, such as classifying an image or detecting anomalies (J. Chen & Ran, 2019; L. Deng et al., 2020b).
- **Optimization:** Utilizing hardware accelerators and optimized software libraries to enhance performance and efficiency (Banbury, Reddi, Lam, Fu, Fazel, Holleman & Whatmough, 2020; Sze, Chen et al., 2017).

Decision Making and Action Based on the analysis, the edge device makes decisions and takes appropriate actions (S. Deng et al., 2020c; Shi et al., 2016e). Actions can include:

- **Autonomous Responses:** Immediate actions taken by the device, such as adjusting a thermostat, triggering an alarm, or controlling a robotic arm (Y.-H. Chen et al., 2017c; B. Cheng et al., 2018b).
- **User Notifications:** Providing feedback or alerts to users, such as health warnings or personalized recommendations (Hoy, 2018c; S. Wang & Krishnan, 2018c).
- **Data Sharing:** Sending summarized insights or critical information to cloud services or other devices for further processing or collaboration (Khemka, 2021; Taleb et al., 2017b).

Feature	Edge AI	Cloud AI	Distributed AI
Processing Location	Local (on edge devices)	Centralized (cloud servers)	Both (edge and cloud)
Latency	Very low	High	Moderate
Bandwidth Usage	Low	High	Moderate
Scalability	Limited by edge device power	High	High
Energy Efficiency	Efficient for low-power tasks	Efficient at scale	Task-dependent
Security & Privacy	High (local data storage)	Lower (data in transit)	Varies (local & cloud)
Application Suitability	Real-time, local decision-making	Large-scale, complex tasks	Mixed workload distribution

Compute distribution structures are well-suited for vastly different use-cases.

Figure 2.2: A comparison of the three kinds of AI

This workflow enables Edge AI systems to operate efficiently, providing timely and context-aware responses without relying heavily on centralized resources.

2.3. Edge AI vs. Cloud AI vs. Distributed AI

2.3.1 Cloud AI

Definition and Overview Cloud Artificial Intelligence (Cloud AI) refers to the delivery of AI services and applications through cloud computing infrastructures (Armbrust et al., 2010b; Q. Zhang et al., 2010). In this model, data is transmitted from end-user devices to centralized data centers where powerful servers perform computationally intensive AI tasks such as training complex models and processing large datasets (Armbrust et al., 2010b; Marston et al., 2011). Cloud AI leverages the scalability, flexibility, and vast computational resources of cloud platforms to provide AI capabilities to users without the need for significant local processing power (Agency, 2021b;

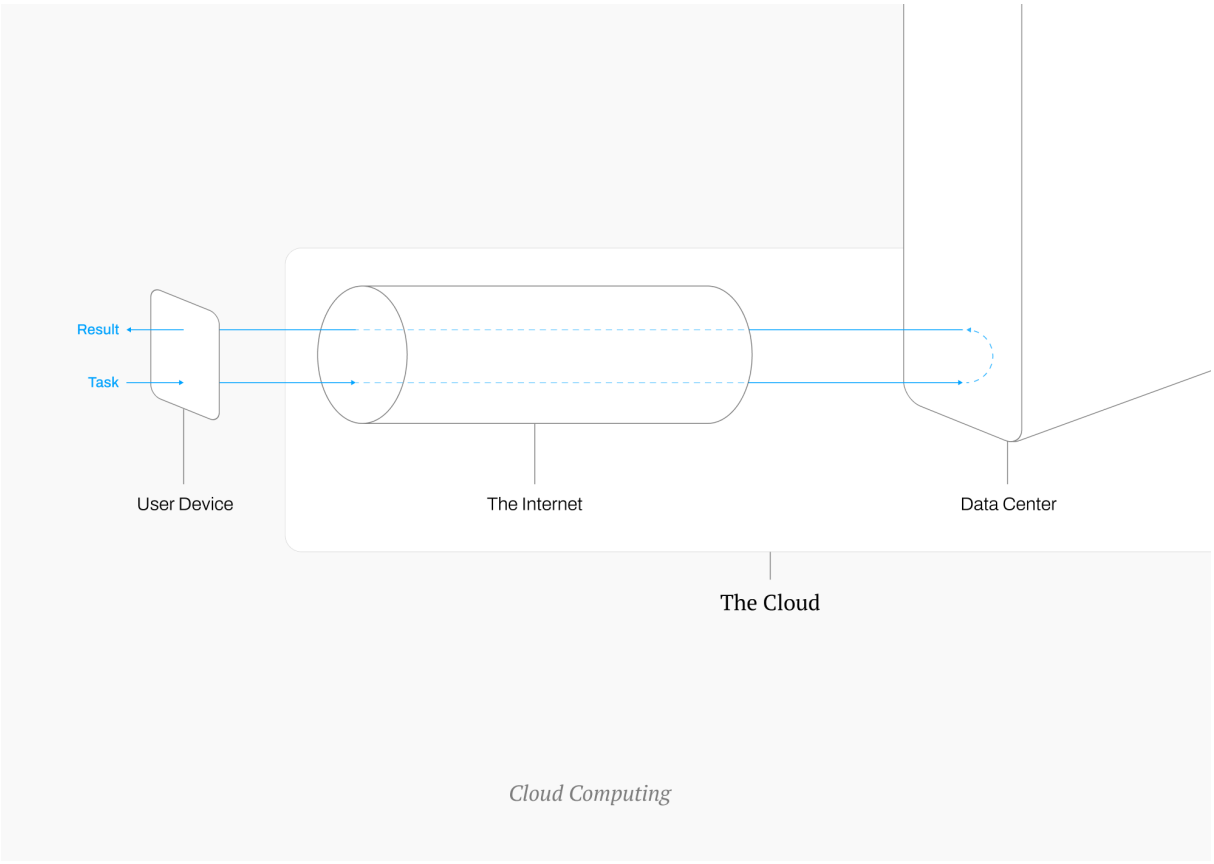

Figure 2.3: The figure showcases computations performed on the cloud

Amazon, 2021).

Cloud computing offers various services, including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS), which are utilized to deploy AI applications (Armbrust et al., 2010b; Q. Zhang et al., 2010). Major cloud providers like Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform offer AI and machine learning services that enable organizations to build, train, and deploy AI models in the cloud (Azure, 2021; Cloud, 2021; Services, 2021).

Key Properties and Limitations

- **Scalability:** Cloud AI provides virtually unlimited computational resources, allowing for the scaling of AI workloads as needed (Armbrust et al., 2010b; Q. Zhang et al., 2010).
- **Accessibility:** Users can access advanced AI tools and services without investing in expensive hardware (Marston et al., 2011; Takabi et al., 2010). This democratizes AI by making it accessible to a broader audience.

- **Centralized Data Processing:** Data from multiple sources is aggregated and processed in centralized data centers (Agency, 2021b; Armbrust et al., 2010b), facilitating comprehensive analytics and model training on large datasets.
- **Managed Services:** Cloud providers offer managed AI services, reducing the complexity of deploying and maintaining AI infrastructure (Azure, 2021; Cloud, 2021; Services, 2021). This allows organizations to focus on application development rather than infrastructure management.

Limitations:

- **Latency:** Transmitting data to and from the cloud introduces latency, which can be detrimental to real-time applications (J. Chen & Ran, 2019; Shi et al., 2016e).
- **Bandwidth Consumption:** Large volumes of data transmission consume significant bandwidth, potentially leading to increased costs and network congestion (**iea2017** ; Systems, 2018b).
- **Privacy and Security Risks:** Sending sensitive data to the cloud raises concerns about data breaches and compliance with privacy regulations like GDPR (Takabi et al., 2010; Union, 2016b). Centralized storage becomes a target for cyberattacks (Abouelmehdi et al., 2018).
- **Dependency on Connectivity:** Cloud AI relies on stable internet connectivity; disruptions can lead to loss of service availability (Y. Mao, You et al., 2017; Shi et al., 2016e).

2.3.2 Distributed AI

Definition and Overview Distributed AI involves the use of multiple interconnected computing nodes working collaboratively to perform AI tasks (Dean & Ghemawat, 2008; H. Tan et al., 2018). This paradigm distributes computational workloads across

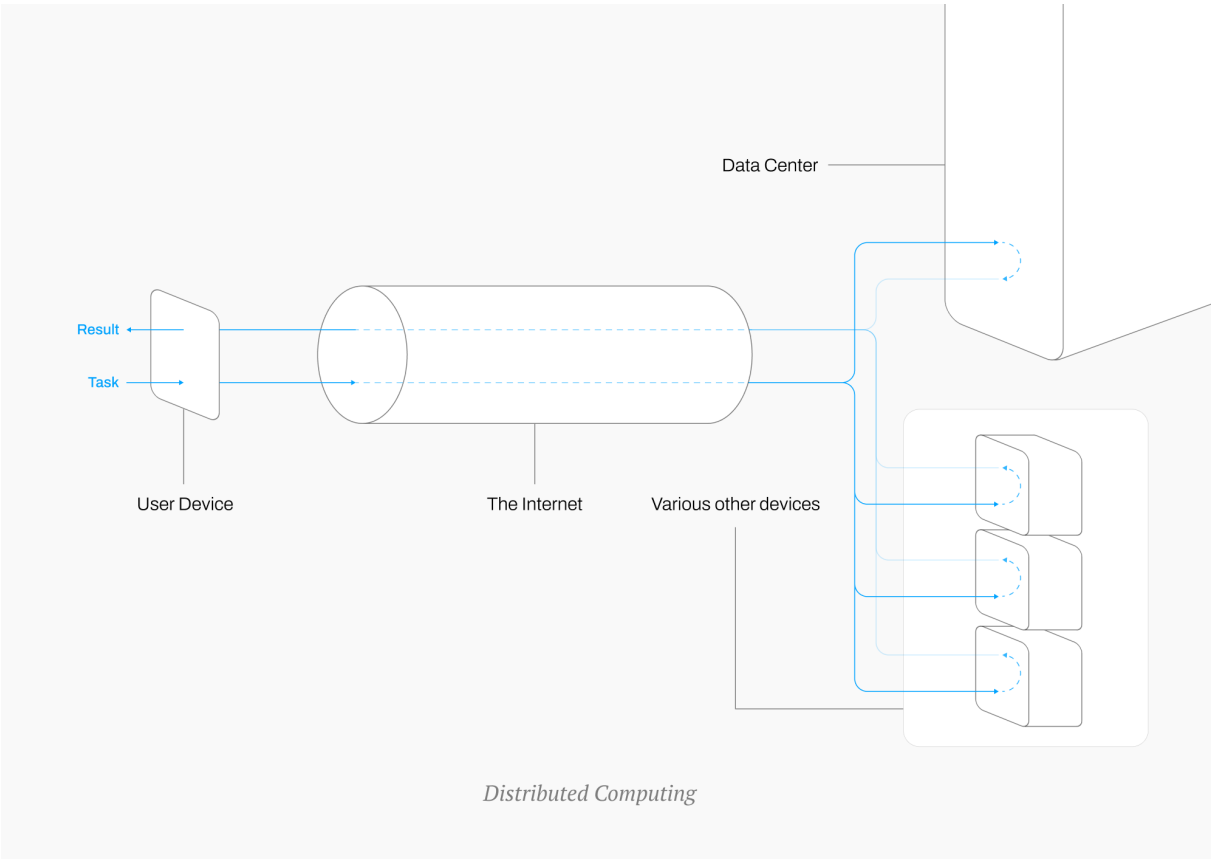

Figure 2.4: Enter Caption

various devices or systems, which may include servers, edge devices, and cloud resources. Distributed AI aims to leverage the combined computational power and data resources of multiple nodes to improve performance, scalability, and fault tolerance (M. Li et al., 2014; Q. Yang, Liu, Chen & Tong, 2019).

Key approaches within Distributed AI include:

- **Distributed Machine Learning:** Training AI models across multiple machines to handle large datasets and reduce training time (Abadi et al., 2016b; Goyal et al., 2017).
- **Federated Learning:** Training models locally on edge devices using local data and aggregating updates centrally, enhancing data privacy (Konečný et al., 2016b; Q. Yang, Liu, Chen & Tong, 2019).
- **Multi-Agent Systems:** Systems where multiple intelligent agents interact or collaborate to solve complex problems (Lyu et al., 2020; Wooldridge, 2009).

Key Properties and Limitations

- **Parallelism:** Distributed AI leverages parallel processing to accelerate computation and handle larger workloads efficiently (Abadi et al., 2016b; H. Tan et al., 2018).
- **Scalability:** Systems can scale horizontally by adding more nodes, accommodating growing data volumes and computational demands (Goyal et al., 2017; M. Li et al., 2014).
- **Data Localization:** By keeping data processing local to each node, distributed AI can enhance privacy and reduce the need for data transmission (Konečný et al., 2016b; Q. Yang, Liu, Chen & Tong, 2019).
- **Fault Tolerance:** The distributed nature allows systems to continue operating even if some nodes fail, enhancing reliability (M. Li et al., 2014; Verbraeken et al., 2020).

Challenges

- **Communication Overhead:** Synchronization and data exchange between nodes can introduce significant communication costs and latency (Konečný et al., 2016b; H. Tan et al., 2018).
- **System Complexity:** Designing and managing distributed AI systems is complex, requiring sophisticated coordination algorithms and infrastructure (Abadi et al., 2016b; Dean & Ghemawat, 2008).
- **Consistency and Convergence:** Ensuring that distributed models converge correctly and consistently is challenging due to potential asynchrony and network delays (J. Chen et al., 2016b; M. Li et al., 2014).
- **Security Risks:** Distributed systems may be vulnerable to attacks targeting individual nodes or communication channels, such as poisoning or adversarial attacks (Lyu et al., 2020; Yin et al., 2018).

2.3.3 Comparative Analysis

Processing Location and Data Flow

- **Edge AI:** Processing occurs locally on edge devices close to the data source, minimizing data transmission (J. Chen & Ran, 2019; Shi et al., 2016e).
- **Cloud AI:** Processing is centralized in remote data centers; data flows from devices to the cloud and back (Agency, 2021b; Armbrust et al., 2010b).
- **Distributed AI:** Processing is shared among multiple nodes, which may include a combination of edge devices, servers, and cloud resources (Dean & Ghemawat, 2008; H. Tan et al., 2018).

Latency and Real-Time Capabilities

- **Edge AI:** Low latency due to on-device processing; ideal for real-time applications requiring immediate responses (J. Chen & Ran, 2019; S. Deng et al., 2020c).
- **Cloud AI:** Higher latency resulting from data transmission delays; less suitable for time-sensitive tasks (Bixby & Renaudin, 2019b; Shi et al., 2016e).
- **Distributed AI:** Latency varies depending on network conditions and synchronization requirements; can be optimized for specific applications (Konečný et al., 2016b; H. Tan et al., 2018).

Bandwidth Usage

- **Edge AI:** Minimal bandwidth usage; reduced dependency on continuous network connectivity (Shi et al., 2016e; Systems, 2018b).
- **Cloud AI:** High bandwidth consumption due to frequent data transmission; requires reliable high-speed connectivity (Systems, 2018b).

- **Distributed AI:** Bandwidth usage varies; communication between distributed nodes can be significant, especially during model synchronization (Konečný et al., 2016b; M. Li et al., 2014).

Privacy and Security Implications

- **Edge AI:** Enhanced privacy as data remains on the device; reduced exposure to network-based security threats (Alrawais et al., 2017b; Omoniwa et al., 2018b).
- **Cloud AI:** Greater risk of data breaches and privacy violations due to centralized data storage (Takabi et al., 2010; Union, 2016b).
- **Distributed AI:** Privacy can be preserved through techniques like federated learning; however, security risks exist in distributed communications (Konečný et al., 2016b; Lyu et al., 2020).

Scalability and Resource Management

- **Edge AI:** Scalability is limited by the capabilities of individual devices; managing a large number of devices can be complex (S. Deng et al., 2020c; Shi et al., 2016e).
- **Cloud AI:** High scalability through elastic cloud resources; resource management is centralized and often automated (Armbrust et al., 2010b; Q. Zhang et al., 2010).
- **Distributed AI:** Scalability is achieved by adding more nodes; however, resource management becomes more complex due to the distributed nature (M. Li et al., 2014; H. Tan et al., 2018).

Application Suitability and Use Cases

- **Edge AI:** Suitable for applications requiring real-time processing, low latency, and enhanced privacy, such as autonomous vehicles (B. Li, Li & Liu, 2018), smart wearables (S. Wang & Krishnan, 2018b), and industrial automation (B. Cheng et al., 2018b).

- **Cloud AI:** Ideal for data-intensive tasks requiring significant computational power, like big data analytics, deep learning model training, and complex simulations (Agency, 2021b; Services, 2021).

- **Distributed AI:** Applicable in scenarios needing collaborative learning and data privacy, such as federated learning for mobile devices (Konečný et al., 2016b), distributed sensor networks (Al-Fuqaha et al., 2015), and large-scale AI model training (Abadi et al., 2016b).

Chapter 3

Thesis for Edge AI

3.1. Introduction

3.1.1 Purpose of the Chapter

The integration of artificial intelligence (AI) into everyday life is accelerating at an unprecedented pace, reshaping industries and redefining human-computer interactions. This chapter aims to explore the transformative potential of Edge AI in enabling ubiquitous intelligence. By examining advancements in hardware, optimization techniques, and the development of decentralized AI networks, we seek to provide a comprehensive understanding of how Edge AI can overcome the limitations of centralized AI infrastructures. The goal is to illustrate how bringing AI computations closer to the data source can revolutionize user experiences, automate mundane decisions, and unlock new applications that were previously unattainable.

3.1.2 The Imperative of Inference Compute and Deployment Speed

As we move toward a future of seamless and pervasive AI adoption, two critical resources emerge as pivotal: **inference compute** and the **speed of deployment**. Inference compute refers to the computational power required for AI models to process new data and generate predictions or actions in real-time. The speed of deploy-

ment pertains to how quickly these computational resources can be made available wherever they are needed. The existing AI infrastructure, heavily reliant on centralized data centers located hundreds of miles away from end-users, presents significant challenges. Latency becomes a pressing issue when data must travel long distances for processing, leading to delays that are unacceptable for applications requiring immediate responses—such as real-time health monitoring or autonomous vehicles. Additionally, as more devices compete for access to centralized resources, bandwidth limitations and network congestion can degrade performance and user experience. Edge AI addresses these challenges by decentralizing computational resources and bringing them closer to the data source—the edge devices themselves. By processing data locally, edge devices can reduce latency, minimize bandwidth usage, and improve efficiency. This proximity enables real-time processing and decision-making, which is crucial for applications that demand instantaneous responses. The importance of inference compute and deployment speed is underscored by the trend toward ubiquitous computing, where computing resources are seamlessly integrated into everyday life. The International Mobile Telecommunications 2030 (IMT-2030, 2020)(IMT-2030) framework, which outlines priorities for future 6G networks, highlights the significance of edge AI in achieving this vision. As AI models become more complex and are integrated into various modalities, the demand for rapid and efficient inference compute will only grow. In essence, making inference compute omnipresent—available wherever you are, whenever you need it—is essential for unlocking the full potential of AI applications. This shift will enable ordinary conversational models to evolve into virtual assistants capable of nuanced understanding, transform health apps into real-time emergency responders, and convert learning tools into personalized tutors that adapt to individual aptitudes.

3.1.3 Overview of the Topics Covered

This chapter is structured to provide a comprehensive exploration of the role of Edge AI in shaping the future of technology and society. The topics covered include:

- **The Need for Ubiquitous Edge AI:** An analysis of the limitations of current AI infrastructures and how Edge AI offers solutions to overcome issues related to latency, bandwidth, and scalability.
- **Advancements in Edge AI Hardware and Techniques:** A review of the latest developments in hardware, such as Qualcomm's Snapdragon processors enabling on-device large language models (LLMs) and NVIDIA's AI-on-5G platform, as well as optimization techniques like model compression, knowledge distillation, and pruning that make Edge AI feasible on lower-resource devices.
- **Decentralized AI Networks and Cross-Platform Training:** An examination of how decentralized networks enable the distribution of computational tasks across various devices, reducing latency and enhancing performance. Discussion on cross-platform training engines that allow AI models to be developed and fine-tuned across diverse hardware environments, ensuring adaptability and scalability.
- **Fundamental Applications Enabled by Edge AI:** An in-depth look at groundbreaking applications made possible by Edge AI.

3.2. The Need for Ubiquitous AI

3.2.1 Limitations of Current AI Infrastructure

Bandwidth and Latency Constraints The current AI infrastructure predominantly relies on cloud-based data centers that are often located far from end-users. This physical distance introduces significant bandwidth and latency challenges. High-bandwidth applications, such as real-time video processing, augmented reality, and interactive gaming, require rapid data transmission between devices and servers. When data has to travel long distances, the latency increases substantially. For applications that demand immediate feedback, such as autonomous vehicles or emergency response systems, even milliseconds of delay

can be critical. Furthermore, the increasing volume of data generated by edge devices exacerbates bandwidth limitations. As devices become more sophisticated and data-intensive, the strain on network resources intensifies. This congestion can lead to slower response times and reduced quality of service, hindering the performance of AI applications that require real-time processing.

Reliance on Centralized Data Centers Centralized data centers have been the backbone of AI processing and storage, providing the computational power necessary for complex tasks. However, this centralized approach has inherent limitations. Dependence on remote servers means that any disruption—be it network failures, maintenance downtime, or physical damages—can halt AI services for users across vast regions. Additionally, as the number of AI-enabled devices grows exponentially, centralized data centers struggle to scale accordingly. This reliance also raises concerns about data privacy and security, as sensitive information must be transmitted and stored in central locations, increasing the risk of breaches.

Issues of Congestion and Efficiency With billions of devices connected to the internet, network congestion has become a significant issue. Centralized infrastructures funnel vast amounts of data through limited channels, leading to bottlenecks. This congestion not only degrades performance but also increases operational costs due to the need for more robust networking equipment and infrastructure. Energy efficiency is another concern; transmitting large volumes of data over long distances consumes considerable energy, contributing to a larger carbon footprint. The inefficiencies inherent in centralized systems hinder the scalability and sustainability of AI applications.

3.2.2 The Vision of Ubiquitous Computing

Definition and Importance Ubiquitous computing, also known as pervasive computing, envisions a world where computing resources are seamlessly integrated into everyday life, accessible anytime and anywhere. In this paradigm, technology becomes an invisible assistant, enhancing human capabilities without explicit interaction. Devices and systems communicate and collaborate autonomously, providing personalized and context-aware services. The importance of ubiquitous computing lies in its potential to transform how we live and work, making technology more intuitive and responsive to human needs.

IMT-2030 Framework and 6G Networks The International Mobile Telecommunications 2030 (IMT-2030) framework outlines the vision and objectives for the future development of mobile networks, commonly referred to as 6G. A key priority of this framework is the integration of AI capabilities at the edge of the network to support the demands of ubiquitous computing (IMT-2030, 2020). 6G networks are expected to offer ultra-low latency, high reliability, and massive connectivity, enabling new applications such as real-time holographic communications, immersive virtual reality, and advanced robotics. Edge AI is instrumental in achieving the goals set forth by the IMT-2030 framework. By decentralizing computational tasks, edge computing reduces the burden on network infrastructure and centralized data centers. This decentralization is crucial for handling the massive data throughput anticipated with 6G networks. Additionally, edge AI enhances the adaptability and responsiveness of services, allowing for dynamic resource allocation and improved quality of experience for end-users. The integration of edge AI with 6G networks supports the development of intelligent, context-aware applications that can operate efficiently in diverse environments. This synergy accelerates the transition toward ubiquitous computing, where seamless connectivity and intelligent processing are the norms rather than the exceptions.

3.3. Advancements in Edge AI Hardware and Techniques

3.3.1 Limitations of Current AI Infrastructure

Qualcomm's Snapdragon Processors for On-Device LLMs The rapid evolution of mobile processors has significantly impacted the feasibility of running complex AI models directly on edge devices. Qualcomm's Snapdragon series has been at the forefront of this advancement, enabling on-device deployment of large language models (LLMs). The latest Snapdragon processors incorporate dedicated AI engines designed to accelerate machine learning tasks, allowing smartphones and tablets to run sophisticated AI applications without relying on cloud services. These processors support on-device inferencing for models like BERT and GPT variants, enabling functionalities such as natural language understanding, voice recognition, and contextual awareness directly on the device (Qualcomm Technologies, Inc., 2023). By processing data locally, these devices reduce latency, enhance privacy by keeping data on-device, and provide real-time responsiveness essential for applications like virtual assistants and personalized user experiences. Qualcomm's initiative to bring LLMs to edge devices is a significant step toward ubiquitous AI. It empowers developers to create applications that can function seamlessly even in environments with limited or no connectivity, ensuring consistent performance and user experience.

NVIDIA's AI-on-5G Platform NVIDIA has introduced the AI-on-5G platform, which integrates AI computing with 5G connectivity to deliver high-performance edge computing solutions. This platform combines NVIDIA's EGX edge computing hardware with 5G virtual radio access networks (vRAN), allowing organizations to deploy AI applications over 5G networks efficiently (NVIDIA Corporation, 2021a). The AI-on-5G platform supports a range of AI work-

loads, including computer vision, natural language processing, and generative AI, providing real-time inferencing capabilities at the edge. By leveraging 5G's low-latency and high-bandwidth characteristics, the platform enables applications such as autonomous robots, smart factories, and immersive AR/VR experiences. The integration of AI and 5G accelerates data processing and decision-making, essential for time-sensitive applications. NVIDIA's platform addresses the growing demand for edge computing solutions that can handle the increasing data generated by IoT devices. It offers a scalable and flexible infrastructure that can adapt to various industry needs, from telecommunications to manufacturing.

3.3.2 Optimization Techniques for Edge AI

To make advanced AI models viable on resource-constrained edge devices, various optimization techniques have been developed. These techniques focus on reducing model size, computational requirements, and energy consumption while maintaining performance.

Model Compression Model compression involves reducing the size of AI models to fit within the limited storage and memory capacities of edge devices. Techniques such as weight quantization convert the model's parameters from high-precision floating-point representations to lower-precision formats like 8-bit integers, significantly reducing the model size and computational overhead (Han et al., 2016a). Another approach is model pruning, where redundant or less significant weights and neurons are removed from the network. This reduces the number of parameters and computations required during inference, enabling faster processing and lower energy consumption on edge devices. Compression techniques are crucial for deploying deep learning models on devices like smartphones, wearables, and IoT sensors, where computational resources are limited. By compressing models without substantial loss in accuracy, developers can bring advanced AI capabilities to a broader range of devices.

Knowledge Distillation Knowledge distillation is a technique where a large, complex model (the teacher) transfers its learned knowledge to a smaller, simpler model (the student). The student model is trained to replicate the outputs of the teacher model, capturing its performance in a more compact form suitable for edge deployment (Hinton et al., 2015a). This approach allows developers to leverage the capabilities of large models while deploying lightweight versions that are efficient enough for real-time inference on edge devices. DistilBERT is an example of a distilled model that retains much of the performance of BERT while being significantly smaller and faster (Sanh et al., 2019a). Knowledge distillation bridges the gap between the need for high-performing models and the limitations of edge hardware. It enables the deployment of sophisticated AI functionalities in environments where computational resources are constrained.

Pruning and Sparse Representations Pruning techniques involve removing unnecessary connections and neurons from neural networks, leading to sparse representations that are more efficient to compute. Structured pruning removes entire neurons or filters, while unstructured pruning removes individual weights. Sparse models require specialized hardware or libraries to exploit the sparsity for computational gains (M. Zhu & Gupta, 2018). Recent advancements have made it feasible to deploy sparse models on edge devices. The PockEngine framework, for instance, enables sparse and efficient fine-tuning of models in resource-constrained environments, making it suitable for edge applications(L. Zhu et al., 2023). PockEngine leverages sparsity to reduce memory footprint and computational load, allowing complex models to run on devices with limited resources. Pruning and sparse representations are vital for extending the capabilities of edge devices. They allow for the deployment of AI models that would otherwise be too large or resource-intensive, enabling a wider range of applications and services.

3.3.3 Enabling Real-Time AI Experiences

The combination of advanced hardware and optimization techniques has paved the way for real-time AI experiences that were previously unattainable on edge devices.

Voice Assistants and Augmented Reality Edge AI enables voice assistants to process speech recognition and natural language understanding locally, providing immediate responses without the need for cloud connectivity. This reduces latency and enhances privacy, as user data does not need to be transmitted over networks(Y. Li et al., 2019). On-device processing allows for continuous operation even in offline scenarios, improving reliability. In augmented reality (AR), edge computing allows for real-time processing of visual data, overlaying digital information onto the physical world with minimal delay. This is crucial for applications like navigation assistance, gaming, and industrial maintenance, where timely and accurate information enhances user experience and productivity. For example, AR glasses powered by edge AI can analyze the environment in real-time, providing users with contextual information, object recognition, and navigation cues. This level of immediacy and responsiveness is essential for immersive experiences.

Personalized Recommendations By processing user data locally, edge AI can generate personalized recommendations in real-time without compromising privacy. Wearable devices can analyze biometric data to provide health and fitness suggestions tailored to the user's current state(S. Patel et al., 2012). Retail applications can leverage edge AI to analyze customer behavior in-store, offering personalized promotions and enhancing the shopping experience. Smart refrigerators and home assistants can suggest meal plans or shopping lists based on user preferences and consumption patterns. Edge AI enables these personalized services to function efficiently without relying on constant cloud connectivity,

ensuring faster response times and greater user satisfaction.

Bridging Performance and Accessibility Edge AI technologies bridge the gap between high-performance AI applications and user accessibility. By enabling complex computations on everyday devices, users can access advanced AI functionalities without the need for expensive hardware or constant internet connectivity. This democratization of AI empowers users in regions with limited infrastructure, allowing them to benefit from AI advancements in education, healthcare, and other critical sectors. Educational apps can provide personalized learning experiences, while health monitoring devices can offer real-time diagnostics. Moreover, edge AI reduces the digital divide by making AI applications more inclusive and widely available. It supports the development of applications that cater to diverse needs, enhancing overall quality of life.

3.4. Decentralized AI Networks and Cross-Platform Training

3.4.1 Decentralized AI Inferencing Networks

Concept and Benefits Decentralized AI inferencing networks involve distributing AI computational tasks across a multitude of edge devices rather than relying solely on centralized cloud servers. In this architecture, each device processes data locally and may share insights or updates with other devices or a central server when necessary (Sajnani & Thilakarathna, 2020). This paradigm shift offers several significant benefits:

– **Reduced Latency:** Local processing minimizes the time delay associated with data transmission to distant servers, enabling real-time responses essential for applications like autonomous driving and emergency services.

- **Bandwidth Efficiency:** By processing data at the source, the amount of data transmitted over the network is reduced, alleviating congestion and lowering operational costs (Shi et al., 2016d).
- **Enhanced Privacy:** Sensitive data can be processed locally without being transmitted over potentially insecure networks, mitigating privacy concerns and complying with data protection regulations.
- **Scalability:** Distributing computational workloads across numerous devices allows the system to handle increasing data volumes without overburdening any single node or central server.

Decentralized inferencing empowers devices with greater autonomy, making AI services more robust and accessible, especially in areas with limited connectivity.

Reducing Latency through Local Processing Latency is a critical factor affecting the performance and user experience of AI applications. In decentralized networks, edge devices perform computations close to the data source, significantly reducing the time it takes to process and respond to information (Satyanarayanan, 2017a). This local processing is particularly vital for time-sensitive applications:

- **Autonomous Vehicles:** Real-time sensor data processing is essential for making instantaneous driving decisions. Local inferencing avoids delays that could be life-threatening (T. Chen et al., 2019).
- **Industrial Automation:** Machinery and robotics require immediate feedback loops to operate safely and efficiently. Edge processing ensures minimal latency in control systems.
- **Healthcare Monitoring:** Wearable devices that track vital signs can alert users or medical professionals instantly if anomalies are detected, enabling prompt interventions (Piwek et al., 2016).

By reducing latency, decentralized AI networks enhance the effectiveness of AI applications across various domains, delivering smoother and more responsive user experiences.

3.4.2 Cross-Platform AI Training Engines

Concept and Benefits Cross-platform AI training engines are frameworks that enable the development, training, and deployment of AI models across diverse hardware platforms and operating systems (Z. Jiang et al., 2021). These engines abstract the complexities of underlying hardware, providing a unified interface for developers. The benefits include:

- **Adaptability:** Models can be easily ported and optimized for different devices, from powerful servers to resource-constrained edge devices.
- **Efficiency:** Cross-platform engines can tailor computational workloads to the specific capabilities of each device, enhancing performance and energy efficiency.
- **Scalability:** Developers can deploy AI solutions widely without rewriting code for each platform, accelerating time-to-market.

Examples of such frameworks include TensorFlow Lite, PyTorch Mobile, and ONNX Runtime, which facilitate seamless model deployment on various devices (TensorFlow Lite, 2023) (PyTorch Mobile, 2023) (ONNX Runtime, 2023).

Reducing Latency through Local Processing Cross-platform AI training engines not only support inference but also enable training and fine-tuning of models directly on edge devices (Lane et al., 2015). This capability is crucial for:

- **Personalization:** Devices can adapt models based on local user data, providing more personalized services without compromising privacy.

- **Continuous Learning:** Edge devices can update models in response to new data, improving performance over time.
- **Offline Operation:** Local training allows devices to function and improve even without network connectivity.

By facilitating on-device training and inference, cross-platform engines reduce dependency on centralized infrastructure and enhance the adaptability of AI applications.

3.4.3 Reducing Dependency on Centralized Cloud Providers

Enhancing Privacy and Security Relying on centralized cloud providers raises concerns about data privacy and security. Transmitting sensitive information over networks exposes it to potential interception and breaches. Decentralized AI networks and on-device processing mitigate these risks by keeping data local (T. Li et al., 2020):

- **Data Sovereignty:** Users maintain control over their data, complying with regulations like GDPR and enhancing trust.
- **Reduced Attack Surface:** Limiting data transmission reduces opportunities for cyberattacks targeting data in transit or at rest on central servers.
- **Secure Processing:** Edge devices can implement hardware-level security measures, such as secure enclaves, to protect data during processing. By enhancing privacy and security, decentralized AI fosters user confidence and meets regulatory requirements.

Empowering Users with Control over Data Decentralized architectures empower users by giving them greater control over their data and how it is used (Truong et al., 2019):

- **Transparency:** Users can more easily understand and manage what data is collected and processed on their devices.

- **Consent Management:** Local control allows users to grant or revoke permissions without relying on external entities.
- **Personalized Experiences:** Users can tailor AI services to their preferences, enhancing satisfaction and engagement.

This empowerment aligns with ethical principles of autonomy and respect for individual rights, fostering a more user-centric approach to AI development.

3.5. Edge AI Automating Unnecessary Choices

3.5.1 The Concept of Cognitive Offloading

Cognitive offloading refers to the process by which individuals use external tools or devices to reduce the mental effort required for certain tasks or decision-making processes (Risko & Gilbert, 2016). This practice allows people to conserve cognitive resources by delegating routine or complex tasks to external aids, such as using a calculator for arithmetic or setting reminders on a smartphone. Cognitive offloading has become increasingly prevalent with the advent of digital technologies, enabling individuals to manage information overload and enhance productivity. The integration of artificial intelligence into everyday devices amplifies the potential for cognitive offloading. AI systems can learn from user behaviors, predict needs, and automate decisions, further reducing the cognitive burden on individuals. This shift allows people to focus on higher-order thinking, creativity, and tasks that require human judgment and emotional intelligence.

3.5.2 Edge AI's Role in Simplifying Daily Decisions

Edge AI plays a pivotal role in advancing cognitive offloading by bringing intelligent processing capabilities directly to the user's environment. By operating on edge devices such as smartphones, wearables, and smart home systems, AI

can provide real-time assistance and automation without the latency or privacy concerns associated with cloud computing (Shi et al., 2016d).

- **Smart Personal Assistants:** Edge AI enables personal assistants to operate locally, understanding user preferences and context to automate routine tasks. For example, scheduling appointments, sending automatic replies, and managing to-do lists can be handled without user intervention (Hoy, 2018a). These assistants can learn habits over time, providing proactive suggestions and handling tasks seamlessly.
- **Home Automation Systems:** Smart homes equipped with edge AI can autonomously control lighting, temperature, security, and appliances based on user patterns and environmental conditions (Alam et al., 2012). By learning from user behaviors, these systems optimize energy consumption and enhance comfort without requiring manual adjustments.
- **Health and Wellness Management:** Wearable devices utilizing edge AI can monitor vital signs, physical activity, and sleep patterns in real-time (Piwek et al., 2016). They provide personalized health recommendations, alert users to potential health issues, and even automate emergency responses if necessary. By continuously analyzing data locally, these devices offer immediate insights while preserving user privacy.
- **Automotive Applications:** Edge AI in vehicles can assist with navigation, monitor driver fatigue, and adjust driving settings based on preferences and conditions (Litman, 2019). By automating these aspects, drivers can focus more on the road and less on ancillary tasks, enhancing safety and convenience.
- **Financial Management:** Edge AI can automate budgeting, bill payments, and investment decisions by analyzing spending habits and financial goals (Klei, 2017). Users receive tailored advice and automated actions that align with their financial objectives, reducing the need for constant monitoring.

By automating routine decisions and tasks, edge AI reduces the mental workload on individuals. This simplification of daily life allows users to allocate their cognitive resources to more meaningful activities, such as personal development, relationships, and creative endeavors.

3.5.3 Impact on Lifestyle and Productivity

The automation of unnecessary choices through edge AI has significant implications for both lifestyle and productivity:

- **Enhanced Efficiency:** Automating routine tasks saves time and reduces errors associated with manual handling (Davenport & Kirby, 2016). Users can accomplish more in less time, improving overall efficiency in personal and professional domains.
- **Improved Decision Quality:** Edge AI systems can process vast amounts of data and recognize patterns that humans might overlook. By providing data-driven recommendations, they enhance the quality of decisions in areas like health, finance, and time management (Silver et al., 2016).
- **Reduced Cognitive Load:** Offloading decisions to AI reduces mental fatigue and decision paralysis, which can occur when individuals are overwhelmed by choices (Vohs et al., 2008). This reduction in cognitive load contributes to better mental health and well-being.
- **Personalization:** Edge AI tailors experiences to individual preferences, providing personalized content, recommendations, and interactions (Adomavicius & Tuzhilin, 2005). This customization enhances user satisfaction and engagement.
- **Accessibility:** Automating tasks makes technology more accessible to individuals with disabilities or limited technical skills. Voice assistants and smart devices can simplify complex operations, promoting inclusivity (Leporini & Paternò, 2008).

However, reliance on edge AI for decision-making also raises considerations:

- **Over-Reliance and Skill Degradation:** Excessive dependence on AI may lead to a decline in critical thinking and problem-solving skills, as individuals become less practiced in making decisions independently (Carr, 2011).
- **Privacy and Security Risks:** While edge AI processes data locally, the collection and storage of personal information still pose privacy risks if devices are compromised (Zheng et al., 2014). Ensuring robust security measures is essential.
- **Ethical Implications:** Automating decisions involves ethical considerations regarding autonomy, consent, and the potential for bias in AI algorithms. Transparency and fairness must be prioritized in AI system design (Danks & London, 2017).
- **Economic Impact:** Automation may affect employment in sectors reliant on routine tasks, necessitating a focus on reskilling and adapting to new job roles (Frey & Osborne, 2017).

In conclusion, edge AI's ability to automate unnecessary choices offers substantial benefits in enhancing lifestyle and productivity. By thoughtfully addressing the associated challenges, society can harness this technology to improve quality of life while fostering personal growth and well-being.

3.6. Fundamental Applications Enabled by Edge AI

3.6.1 Hyper-Personalized Learning Assistants

Real-Time Knowledge Translation Edge AI empowers the development of hyper-personalized learning assistants that can provide real-time translation of

complex technical jargon into language that aligns with the user's existing knowledge base (Khosravi & Cooper, 2018). By processing data locally on the device, these assistants can instantly adapt educational content to the learner's proficiency level without the latency associated with cloud processing. For example, a student studying advanced physics can receive explanations that relate new concepts to their understanding of mathematics, making learning more intuitive and effective. This idea was originally proposed by Sven Wellmann (Polychain).

Adaptive Learning Planners Adaptive learning planners utilize edge AI to analyze a user's learning patterns, strengths, and weaknesses in real-time (Pardo & Siemens, 2014). By continuously monitoring progress, these planners can adjust curricula, suggest relevant resources, and set personalized goals. The local processing of personal learning data enhances privacy while enabling a tailored educational experience that can adapt to changes in the user's aptitude across different subjects.

3.6.2 Live Automated Customer Service

Empathetic AI Agents Edge AI facilitates the creation of empathetic customer service agents capable of understanding and responding to customer emotions and concerns with human-like empathy (Picard, 2003). These agents analyze vocal tones, language nuances, and contextual cues locally to provide personalized support. By operating on the edge, they reduce response times and enhance data security, leading to more satisfying customer interactions.

Eliminating Wait Times Traditional customer service often involves lengthy wait times, leading to customer dissatisfaction. Edge AI agents can handle inquiries instantly by processing requests directly on user devices or local servers (Davenport & Ronanki, 2018). This immediate response capability eliminates queues, providing customers with prompt assistance and freeing up human rep-

resentatives to handle more complex issues.

3.6.3 Sensory Augmentation and Substitution

Wearable Edge AI Devices Wearable devices equipped with edge AI can augment or substitute human senses, offering new levels of accessibility and interaction (Cassinelli & Ishikawa, 2005). For instance, smart glasses for the visually impaired can interpret visual information and convert it into audio descriptions in real-time. By processing data on-device, these wearables maintain user privacy and function effectively without relying on constant internet connectivity.

Enhancing Human Senses Beyond aiding those with impairments, edge AI wearables can enhance normal sensory experiences (Mann, 2014). Devices can amplify hearing in noisy environments, enhance vision in low-light conditions, or provide haptic feedback in virtual reality applications. This sensory augmentation opens up new possibilities for human-computer interaction and immersive experiences.

3.6.4 Immersive Digital Twins

Real-Time IoT Integration Edge AI enables the creation of immersive digital twins—virtual replicas of physical environments that update in real-time based on data from IoT sensors (Tao et al., 2019). By processing sensor data locally, these digital twins provide accurate and up-to-date representations without the delays of cloud processing. Users can interact with these environments through augmented reality (AR) or virtual reality (VR) interfaces for simulations, training, or monitoring.

Applications in Smart Homes and Cities In smart homes, digital twins allow residents to visualize and control home systems like lighting, heating, and security through an interactive virtual model (Alam et al., 2012). In urban

planning, cities can use digital twins to simulate traffic flow, infrastructure development, and environmental impacts, facilitating data-driven decision-making and efficient resource management (Batty, 2018).

3.6.5 Precision Agriculture with AI

Agricultural Drones and Robotics Edge AI powers drones and robotic systems that perform precision agriculture tasks such as targeted irrigation, fertilization, and pest control (C. Zhang et al., 2019). By analyzing data from soil sensors and weather conditions locally, these systems optimize resource usage and improve crop yields. Real-time processing ensures that adjustments are made promptly, responding to changing environmental factors.

Automated and Efficient Food Production Edge AI enables automated monitoring and management of agricultural processes in vertical farms and greenhouses (Benke & Tomkins, 2017). Systems can control lighting, temperature, and nutrient delivery with high precision, maximizing growth rates and conserving resources. Continuous local data analysis allows for immediate adjustments, leading to more efficient and sustainable food production.

3.6.6 Seamless Brain-Computer Interfaces

Neural Interfaces Powered by Edge AI Edge AI facilitates the development of brain-computer interfaces (BCIs) that translate neural signals into commands for devices (Nicolas-Alonso & Gomez-Gil, 2012). By processing neural data locally, BCIs can operate with low latency, providing real-time interaction between the user's brain and external systems. Applications include controlling prosthetic limbs, communicating for individuals with speech impairments, and interacting with virtual environments.

Merging Mind and Machine The integration of edge AI with BCIs blurs the line between human cognition and machine processing (Lebedev & Nicolelis, 2017). Users can experience seamless interaction with technology, accessing information, or controlling devices through thought alone. This merging of mind and machine opens new frontiers in human capabilities and personalized computing experiences.

3.6.7 Autonomous Vehicles with On-Board Edge AI

Local Sensor Data Processing Autonomous vehicles rely on edge AI to process vast amounts of sensor data, including lidar, radar, and cameras, directly on-board (Badue et al., 2021a). Local processing enables real-time decision-making critical for safe navigation, obstacle avoidance, and adherence to traffic laws. By reducing dependency on external networks, vehicles maintain functionality even in areas with poor connectivity.

Enhancing Safety and Reliability Edge AI enhances the safety and reliability of autonomous vehicles by providing consistent performance without latency (P. Lin et al., 2011). Immediate processing of environmental data allows for quick reactions to dynamic road conditions, reducing the risk of accidents. Redundancy systems can also be implemented to ensure continuous operation in case of hardware failures.

3.6.8 Hive Minds and AI Collaboratives

Collective Intelligence Networks Edge AI enables the formation of collective intelligence networks where multiple devices and AI agents collaborate to solve complex problems (Woolley & Malone, 2011). By sharing insights and processing tasks locally, these networks can tackle challenges that exceed the capabilities of individual systems. Applications range from coordinated disaster response to large-scale environmental monitoring.

Solving Complex Global Challenges Collaborative edge AI networks can address global issues such as climate change, resource management, and disease outbreaks (Helbing, 2019). By aggregating data and processing power from distributed sources, these networks facilitate comprehensive analysis and coordinated action, leveraging collective intelligence for the greater good.

3.6.9 Emotional AI Companions

Photorealistic AI Characters Edge AI allows for the creation of photorealistic AI companions that interact with users in natural and emotionally intelligent ways (McDuff & Czerwinski, 2018). By processing facial expressions, speech patterns, and contextual cues locally, these companions provide personalized interactions that adapt over time. Applications include therapy support, companionship for the elderly, and educational tutoring.

Building Long-Term Relationships Emotional AI companions can form long-term relationships with users by learning from past interactions and evolving their personalities (Bickmore & Picard, 2005). Edge processing ensures that personal data remains secure, fostering trust and deeper engagement. These companions can provide consistent support, enhancing mental well-being and social connectivity.

3.6.10 AI-Generated Pocket Universes

Real-Time Virtual World Creation Edge AI enables the generation of personalized virtual environments, or "pocket universes," that users can explore and manipulate in real-time (Ritchie & Thomas, 2015). By processing user inputs and environmental data locally, these virtual worlds offer immersive experiences tailored to individual preferences. Applications include gaming, virtual tourism, and creative expression.

Exploration and Manipulation of Environments Users can interact with AI-generated worlds through AR and VR devices, altering landscapes, creating objects, and sharing experiences with others (Anthes et al., 2016). Edge AI provides the computational power needed for complex simulations without reliance on external servers, enhancing responsiveness and personalization.

3.6.11 Massively Multiplayer Mixed Reality

Persistent AR Worlds Edge AI supports persistent augmented reality worlds where multiple users can interact with digital content overlaid on the physical environment (Billinghurst et al., 2015). By processing data locally, these experiences are more responsive and can function in areas with limited connectivity. Applications include collaborative workspaces, social platforms, and interactive entertainment.

Shared Experiences and Collaboration Massively multiplayer mixed reality allows users to collaborate on projects, play games, or attend virtual events in shared spaces (Janssen et al., 2019). Edge AI ensures synchronization and real-time interaction, enhancing the sense of presence and community.

3.6.12 Hyperlocal Weather Control

Precision Weather Forecasting Edge AI processes data from dense networks of local sensors to provide high-resolution weather forecasts (Schulz & Mayer, 2018). By analyzing atmospheric conditions in real-time, these systems offer precise predictions for specific locations, aiding in agriculture, event planning, and disaster preparedness.

Localized Weather Interventions Advanced applications involve using edge AI to control weather conditions on a micro-scale, such as dispersing fog at airports or inducing rain over drought-stricken areas (Rosenfeld et al., 2010).

While still largely theoretical, these interventions could optimize environmental conditions for various human activities.

3.6.13 Adaptive Smart Cities

Real-Time Urban Management Edge AI enables smart cities to manage resources like energy, water, and transportation systems in real-time (Khatoun & Zeadally, 2016). By processing data from IoT devices locally, cities can respond immediately to changing conditions, such as adjusting traffic signals to alleviate congestion or rerouting power during outages.

Optimization of Utilities and Services Adaptive systems can optimize utility usage based on demand patterns, reducing waste and costs (Zanella et al., 2014). Services like waste management, public safety, and environmental monitoring benefit from edge AI's ability to analyze data quickly and implement solutions efficiently.

3.6.14 AI-Assisted Creativity Tools

Collaborative Artistic Endeavors Edge AI provides tools for artists, writers, and musicians to collaborate with AI in real-time (Davis et al., 2015). By generating suggestions, enhancing creativity, and automating routine tasks, these tools augment human creativity. Local processing ensures immediate feedback and preserves the originality of the creative process.

Personalized Feedback and Adaptation Creative applications can adapt to the user's style and preferences, offering personalized guidance and inspiration (Lubart, 2005). Whether composing music or designing graphics, AI-assisted tools enhance productivity.

3.6.15 AI-Powered Personal Shoppers

Customized Shopping Experiences Edge AI personal shoppers analyze user preferences, shopping habits, and style to provide tailored product recommendations (Xiao & Benbasat, 2007). By processing data locally, they maintain privacy while delivering highly relevant suggestions across various retail platforms.

Automated Negotiation and Recommendations These assistants can automate price comparisons, negotiate deals, and manage orders on behalf of the user (Maes, 1994). Edge processing ensures quick responses and secure handling of financial information.

3.6.16 Personalized Health Monitoring and Early Disease Detection

Continuous Vital Sign Monitoring Wearable devices with edge AI capabilities continuously monitor vital signs such as heart rate, blood pressure, and glucose levels (Ching et al., 2018). By analyzing data in real-time, they can detect anomalies and alert users or healthcare providers immediately.

Predictive Health Analytics Edge AI enables predictive analytics that identify potential health issues before they become critical (Esteva et al., 2019). By recognizing patterns and trends in physiological data, these systems support proactive healthcare management, leading to better outcomes and reduced medical costs.

3.7. Future Trends and Research Directions

3.7.1 Advancements in Edge AI Technologies

Next-Generation Hardware Innovations The evolution of Edge AI is heavily dependent on advancements in hardware that can support complex AI compu-

tations while adhering to the constraints of edge environments, such as limited power and space. Next-generation hardware innovations are focusing on specialized processors and architectures designed to enhance performance and energy efficiency.

- **Neuromorphic Computing** Neuromorphic computing aims to mimic the neural structure and functioning of the human brain to achieve higher efficiency in processing AI tasks. Companies like Intel and IBM are developing neuromorphic chips that use spiking neural networks to process information more efficiently than traditional architectures (Davies et al., 2018a). These chips have the potential to revolutionize Edge AI by enabling real-time processing of sensory data with minimal power consumption.
- **Quantum Computing** Quantum computing, although still in its nascent stages, holds promise for dramatically accelerating AI computations. Integrating quantum processors into edge devices could potentially solve complex optimization problems and handle large datasets more efficiently (Schuld et al., 2015). Research is ongoing to make quantum computing more accessible and practical for edge applications.
- **Advanced AI Accelerators** Specialized AI accelerators, such as Google's Edge TPU and NVIDIA's Jetson series, are designed to optimize machine learning tasks on edge devices (Google, 2023)(NVIDIA Corporation, 2023). These accelerators offer high performance with low power consumption, enabling more sophisticated AI applications in areas like computer vision, natural language processing, and autonomous navigation.
- **3D Integrated Circuits** Advancements in 3D integrated circuits (ICs) allow for stacking multiple layers of components, reducing the physical footprint and improving performance (Lim et al., 2012). This technology enables more powerful and compact edge devices, facilitating the deployment of AI in space-constrained environments like wearable technology and IoT sensors.

Emerging Software and Algorithms The development of new software frameworks and algorithms is crucial for optimizing AI performance on edge devices.

- **Lightweight Neural Networks** Designing lightweight neural network architectures, such as MobileNetV3 and EfficientNet, is essential for running AI models on resource-constrained devices (A. Howard et al., 2019a). These networks reduce computational complexity and memory usage without significantly compromising accuracy, making them ideal for edge deployment.
- **AutoML and Neural Architecture Search (NAS)** AutoML and NAS automate the process of designing neural networks optimized for specific tasks and hardware constraints (Elsken et al., 2019). These techniques can generate models tailored for edge devices, balancing performance and efficiency, and reducing the need for extensive human expertise in model design.
- **Federated Learning Enhancements** Advancements in federated learning algorithms are improving the efficiency and scalability of decentralized training (T. Li et al., 2020). Techniques to handle non-IID (Independent and Identically Distributed) data, address communication bottlenecks, and ensure robust aggregation are making federated learning more practical for widespread edge deployment.
- **Privacy-Preserving Algorithms** Developing algorithms that protect user data while performing AI computations is a growing area of research (Dwork & Roth, 2014). Techniques like differential privacy, homomorphic encryption, and secure multi-party computation are being adapted for edge environments to ensure data security without sacrificing performance.

3.7.2 Integration with Emerging Technologies

Next-Generation Hardware Innovations Integration with emerging technologies requires hardware innovations that can support new functionalities and

interconnectivity.

- **5G and 6G Networks** The rollout of 5G networks and the development of 6G technology are pivotal for the future of Edge AI (C. Zhang et al., 2019). These networks provide high bandwidth, low latency, and massive device connectivity, enabling real-time data processing and communication between edge devices and cloud infrastructure.
- **IoT Device Integration** Edge AI is becoming increasingly intertwined with the Internet of Things (IoT), necessitating hardware that can seamlessly integrate AI capabilities into a wide array of devices (Stojkoska & Trivodaliev, 2017). Advances in microprocessors and sensors allow for the embedding of AI functions directly into IoT devices, enhancing their autonomy and intelligence.
- **Energy-Efficient Hardware** Research into new materials and designs for energy-efficient hardware is critical for sustaining edge devices (D. Seo et al., 2016). Innovations like graphene-based transistors and ultra-low-power chips enable longer battery life and reduce the environmental impact of widespread edge deployment.

Emerging Software and Algorithms Software developments are essential for integrating Edge AI with other emerging technologies effectively.

- **Edge-Oriented Middleware** Middleware solutions are being developed to manage the complexity of edge environments, providing abstraction layers that simplify development and deployment (Bonomi et al., 2012). These platforms facilitate communication between heterogeneous devices and support scalability and interoperability.
- **AI for Network Optimization** Applying AI to optimize network operations, such as traffic management and resource allocation, enhances the performance of edge networks (N. Zhang et al., 2018). Machine learning al-

gorithms can predict network conditions and adjust parameters in real-time, improving reliability and efficiency.

- **Cross-Domain AI Models** Developing AI models that can operate across different domains and data modalities is a growing focus (Baltrusaitis et al., 2019). These models enable more holistic applications, such as combining visual, auditory, and contextual data for more accurate and versatile AI systems.

Chapter 4

Why does Edge AI need crypto?

4.1. Introduction

4.1.1 Purpose of the Chapter

The crypto and AI communities often don't look each other eye-to-eye. This chapter is an attempt to showcase the usefulness of crypto's fundamental primitives to solve the problems pertinent to edge AI. The purpose of this chapter is to explore how crypto and blockchain can address the challenges in edge AI. By integrating crypto into edge AI systems, we aim to demonstrate how these technologies can intertwine to enhance security, ensure data integrity, incentivize participation, and unlock new potentials for deploying Large Language Models (LLMs) and other AI applications on edge devices.

4.1.2 The Intersection of Edge AI and Crypto

The convergence of edge AI and crypto represents a synergistic blend of decentralized computing and secure, trustless transactions. Edge AI decentralizes processing and computation by distributing computational tasks to the network's periphery, reducing dependence on centralized cloud services. The premise of edge AI is a very one to that of crypto, i.e, the decentralization of trust to the

network's periphery, eliminating the need for central authorities. This culmination of edge AI and crypto addresses several critical aspects:

- **Decentralization and Trust:** In a decentralized network of edge devices, establishing trust without central oversight is challenging. Trust in crypto and blockchains is derived mathematically; computational and mathematical trust are essential for trustless interactions — this is a property that AI currently lacks.
- **Resource Allocation and Incentivization:** Deploying and maintaining edge networks requires substantial resources. Crypto-economic models/-tokens can incentivize individuals and organizations to contribute computational power, data, and other resources by offering token-based incentives.

By leveraging crypto, AI can overcome its inherent limitations, leading to more robust, secure, and efficient systems capable of supporting sophisticated AI applications directly on edge devices.

4.1.3 Overview of Topics Covered

This chapter provides a comprehensive exploration of how crypto technologies can meet the needs of Edge AI. The topics are organized as follows:

- **The Need for Crypto in Edge AI:** We begin by discussing the specific challenges faced by Edge AI, including security vulnerabilities, privacy concerns, trust issues in decentralized environments, high capital expenditure (CapEx) requirements, and the need for effective incentive mechanisms.
- **Leveraging Blockchain for Edge AI:** This section delves into how blockchain technology can enhance security and decentralization in Edge AI networks. We explore mechanisms for secure data sharing, the role of decentralized physical infrastructure networks (DePIN) and TEE networks in reducing CapEx, and case studies like the Helium Network that exemplify these concepts.

- **Advanced Cryptographic Techniques for Edge AI:** As we move forward, we expect to see Trusted Execution Environments (TEEs) be used in building edge AI networks. We also expect to see the use of Multi-Party Computation (MPC) for AI.
- **Decentralized Finance (DeFi) Models for Edge AI Resource Allocation:** This section explores how DeFi concepts like staking, lending, and liquidity pools can optimize resource allocation in Edge AI networks. We discuss the economic incentives that encourage participation and resource sharing.
- **Scaling Solutions for Edge Devices:** We address the scalability challenges by introducing Layer 2 (L2) solutions. These scaling methods reduce computational and bandwidth requirements, making blockchain integration feasible for edge devices.
- **Federated Learning and Blockchain Integration:** We explore the integration of federated learning with blockchain technology to enhance data privacy, security, and incentivization in collaborative AI model training.
- **Challenges and Future Directions:** We identify the technical, operational, and regulatory challenges inherent in integrating crypto with Edge AI. We also highlight future trends, such as advances in cryptographic protocols and the integration with emerging technologies like 5G/6G networks and the Internet of Things (IoT).

Through this structured exploration, we aim to provide valuable insights into why Edge AI needs crypto and how this integration can drive innovation, efficiency, and security in decentralized AI applications deployed on edge devices.

4.2. The Need for Crypto in Edge AI

4.2.1 Decentralization and Trust Issues

In a decentralized edge network comprising numerous devices from different manufacturers and owners, establishing trust and ensuring secure interactions pose significant challenges.

- **Lack of Central Authority:** Traditional centralized authentication mechanisms are not suitable for decentralized networks. Without a central authority to verify identities and enforce policies, it's difficult to ensure all devices are trustworthy (Nguyen et al., 2021).
- **Authentication and Authorization:** Verifying the identity of devices and granting appropriate access rights are critical to prevent unauthorized actions within the network. Edge devices need a secure method to authenticate each other without relying on centralized systems (T. Li et al., 2020).
- **Data Integrity and Tamper Resistance:** Ensuring that the data exchanged between devices has not been tampered with is essential for the reliability of AI models and applications. Maintaining data integrity in decentralized environments is complex (Y. Wang & Su, 2019).
- **Trust in Data Sources:** Devices may be reluctant to trust data or updates from unknown or unverified sources, hindering collaboration and data sharing within the network (Gupta & Tanwar, 2021).

4.2.2 High Capital Expenditure (CapEx) in Edge AI Deployment

Deploying Edge AI infrastructure on a large scale requires substantial financial investment, which can be a barrier to entry for many organizations.

- **Infrastructure Costs:** Building and maintaining the physical infrastructure—including manufacturing edge devices, setting up communication networks, and developing software ecosystems—entails significant upfront costs (Brown et al., 2020).
- **Scalability Limitations:** High CapEx can limit the scalability of Edge AI deployments. Only organizations with substantial financial resources can invest in the necessary infrastructure, leading to slower adoption rates and limited coverage (W. Lin & Wang, 2019).
- **Resource Underutilization:** Traditional deployment models may result in underutilized resources, as dedicated infrastructure might not always operate at full capacity, leading to higher operational costs (N. Zhang et al., 2018).
- **Economic Barriers for Small Entities:** Smaller companies and startups may struggle to compete with larger corporations due to the high costs associated with deploying and maintaining Edge AI networks (Ahmad & Lee, 2020). The best aspect of blockchains is their ability to propagate shared economies of scale. when workloads are distributed they should in theory reduce costs of the core business which comes back to users in the form of savings.
- **Operational and Legal Costs:** Maintaining Edge AI infrastructure incurs ongoing operational expenses, such as energy consumption, maintenance, and skilled personnel, while legal costs arise from ensuring compliance with data privacy and regulatory requirements.

4.2.3 Incentive Mechanisms for Edge AI Networks

Creating sustainable and efficient Edge AI networks requires mechanisms that incentivize participation and resource sharing among diverse stakeholders.

- **Motivating Participation:** Without proper incentives, device owners and

organizations may be reluctant to contribute computational resources, data, or infrastructure. Concerns about costs, privacy, and security can deter participation (E. K. Lee & Lee, 2019).

- **Resource Sharing:** Encouraging the sharing of underutilized computational power, storage, and data can enhance the network's efficiency and scalability. Participants need fair compensation for their contributions (Feng & Zhang, 2018).
- **Economic Models for Collaboration:** Developing economic models that reward contributors appropriately is essential. These models should align the interests of all participants, ensuring optimal network operation (Shi et al., 2016a).
- **Preventing Free-Riding:** Without effective incentive mechanisms, some participants may benefit from the network's resources without contributing their fair share, leading to imbalances and potential degradation of service quality (Huang & Li, 2020).
- **Tokenization and Rewards:** Introducing token-based reward systems can provide tangible incentives. Tokens can represent value and be exchanged or traded, creating a dynamic economy within the Edge AI network (T. Li & Ma, 2021). The flipside of incentivisation systems through tokens is that actors can be incentivized to lie; therefore, verifiability/provability is a property that needs to be paid attention to.

4.3. Leveraging Blockchain for Edge AI

4.3.1 Blockchain for Secure, Decentralized Data Sharing

Blockchain facilitates secure data sharing among edge devices without the need for centralized intermediaries.

Data Integrity and Tamper Resistance

Blockchain provides an immutable ledger where transactions are recorded in blocks linked via cryptographic hashes. This ensures that once data is recorded, it cannot be altered without consensus from the network participants. In the context of edge AI, this immutability guarantees that data collected and shared among devices remains trustworthy and unaltered (Goldreich, 2009). This is the whole premise behind decentralized data storage for uptime and censorship resistance. For example, when edge devices contribute data to train a machine learning model, recording the data transactions on the blockchain prevents malicious actors from injecting false data or modifying existing data.

Decentralized Data Exchange Mechanisms

Blockchain enables decentralized data exchange among edge devices, eliminating the need for a central authority to manage data sharing. Each device can independently validate transactions and share data with others in a peer-to-peer network, ensuring transparency and reducing single points of failure. For example, in the financial sector, decentralized data exchange can facilitate real-time sharing of market data, transaction records, or credit risk assessments among various financial institutions. By using blockchain, banks, investment firms, and payment processors can independently validate and share data such as transaction histories or loan agreements, ensuring that all parties have access to accurate and consistent information.

Privacy-Preserving Data Sharing

Privacy concerns are paramount in edge AI applications, especially when dealing with sensitive personal data. Cryptographic methods such as Homomorphic Encryption (HE) and Fully Homomorphic Encryption (FHE) allow edge devices to verify the authenticity and integrity of data without accessing the raw data it-

self. The other mechanism to securely process data is by using Trusted Execution Environments (TEEs).

4.3.2 Decentralized Physical Infrastructure Networks (DePIN)

Reducing CapEx with Crypto Incentives

Traditional infrastructure deployment requires significant capital expenditure (CapEx), often limiting large-scale deployments to well-funded entities. DePIN leverages crypto incentives to crowdsource the deployment and maintenance of physical infrastructure. Participants are rewarded with tokens for contributing resources such as hardware devices, computational power, or connectivity services.

This model reduces the upfront costs for infrastructure deployment by distributing them across a network of participants motivated by potential financial returns.

Case Study: Helium Network

Helium creates a decentralized wireless network for IoT devices. Individuals and businesses purchase and deploy Helium hotspots, which provide network coverage and, in return, earn tokens as rewards. By incentivizing users to contribute to network infrastructure, Helium has achieved rapid expansion with over 1 million hotspots deployed globally within a few years. This was accomplished at a fraction of the cost required for traditional network deployments by telecommunications companies. The other notable examples of DePIN networks for the internet are: Dawn Network (by the Andrena team) and Roam Network.

Other DePIN Examples

WeatherXM leverages blockchain to build a decentralized network of weather stations. Participants deploy weather stations that collect atmospheric data, which is then shared on the network. Contributors are rewarded with tokens for providing accurate and reliable data, which can be used for weather forecasting, climate research, and agricultural planning.

4.3.3 Incentivizing Data Collection from Edge Devices

Data is a critical asset for training AI models, and collecting high-quality data from edge devices can be challenging. Crypto-economic incentives provide a mechanism to encourage data sharing and participation.

Token Incentives for Data Contributors

By rewarding data contributors with tokens, edge AI networks can motivate individuals and organizations to share their data. This approach creates a decentralized data economy where contributors are compensated for their efforts, and data consumers gain access to valuable datasets.

Case Studies

Hivemapper Hivemapper is a decentralized mapping platform that incentivizes users to capture and upload geospatial data. Participants use dashcams or drones to record imagery of streets and landscapes, contributing to a global, up-to-date map. In return, they earn tokens based on the quantity and quality of the data provided. This model accelerates map creation and updates, outperforming traditional centralized mapping services in coverage and freshness.

DIMO DIMO is a decentralized platform for collecting and sharing vehicle data. Drivers install a device in their cars that collects data on vehicle perform-

ance, location, and usage patterns. Participants are rewarded with tokens for contributing data, which can be used for applications such as improving vehicle diagnostics, enhancing ride-sharing services, or developing autonomous driving algorithms.

Nodle Network The Nodle Network leverages smartphones as edge devices to create a decentralized IoT network. Users install the Nodle app, which utilizes Bluetooth connectivity to collect data from nearby IoT devices and sensors. Participants earn tokens for contributing to network coverage and data collection. This approach transforms smartphones into nodes that support IoT connectivity and data aggregation without the need for additional hardware.

Data DAOs and Decentralized Data Marketplaces

Data Decentralized Autonomous Organizations (Data DAOs) are collective entities governed by smart contracts that facilitate the pooling, management, and monetization of data. They enable participants to contribute data, participate in decision-making, and share in the economic benefits generated. Data DAOs promote transparency, fairness, and community governance in data management. They can be used to create decentralized data marketplaces where data providers and consumers transact directly, ensuring that data contributors are fairly compensated and that data usage complies with agreed-upon terms. Vana is pioneering the Data DAO model by enabling individuals to own and control their personal data. Participants contribute data from various sources, such as health metrics or IoT devices, and collectively decide how the data is used and monetized. This empowers users, promotes privacy, and fosters the development of AI applications that rely on diverse, high-quality datasets. Another example project is Grass. Grass is a decentralized data platform that aims to redefine internet incentive structures by rewarding users for contributing their unused internet bandwidth. Participants run Grass nodes, sharing their surplus

	Traditional PoW (Proof-of-Work)	PoUW (Proof-of-Useful-Work)
Purpose of computation	Miners solve arbitrary cryptographic puzzles (e.g. SHA-256 hash function) that have no value beyond securing the blockchain	Miners perform computations that have external value, such as AI model training, data analysis, or other computational tasks beneficial to various applications.
Resource Utilization	High energy consumption with no additional benefits.	Energy and computational resources contribute to useful outcomes, improving overall system efficiency.
Incentive Structures	Rewards miners solely for securing the network.	Rewards miners (edge devices) for both securing the network and contributing valuable computational work.

Comparative analysis of blockchain verification systems.

Figure 4.1: Traditional Proof of Work Systems vs Proof of Useful Work

bandwidth with the network to earn rewards and support the growth of AI. The platform's Sovereign Data Rollup leverages a network of nodes, routers, validators, zero-knowledge processors, and a data ledger to facilitate data sourcing and transformation—converting unstructured web data into structured datasets. This model enhances data availability and processing efficiency, outperforming traditional centralized data services in scalability and effectiveness.

4.4. Advanced Cryptographic Techniques for Edge AI

4.4.1 Proof-of-Useful-Work (PoUW)

Applying PoUW in Edge AI Networks

Proof-of-Useful-Work (PoUW) aims to redefine this approach by ensuring that the computational effort expended during the consensus process is directed towards tasks with intrinsic value, such as training machine learning models, processing data, or performing scientific computations. In the context of Edge AI, where computational resources are distributed across numerous edge devices like smartphones, IoT sensors, and embedded systems, PoUW presents an innovative solution to harness these dispersed resources effectively. By integrating PoUW into edge networks, devices can participate in securing the blockchain while simultaneously contributing to collective AI tasks, optimizing resource utilization, and creating new incentive structures. This approach not only enhances network security but also provides practical benefits by utilizing computational power for meaningful purposes. This has long been spoken about but we are yet to see meaningful implementations of PoUW in the edge AI space.

Distributed AI Model Training via PoUW

Edge devices often have idle computational capacities that can be utilized for training machine learning models. In a PoUW system:

- **Task Allocation:** The network distributes segments of AI training tasks to participating edge devices.
- **Consensus Mechanism:** Devices perform these tasks as part of the consensus process.

- **Validation:** Other nodes verify the correctness of computations using verification algorithms or spot-checking techniques.
- **Reward Distribution:** Devices receive cryptocurrency tokens as rewards for their useful work, incentivizing continued participation. This model leverages the collective computational power of edge devices, enabling the training of large-scale AI models without the need for centralized data centers.

Federated Learning Integration In federated learning, models are trained across multiple decentralized devices holding local data samples without exchanging them. PoUW can enhance this by:

- **Secure Aggregation:** Using blockchain to securely aggregate model updates from edge devices.
- **Incentivization:** Providing tokens to devices that contribute to model training, encouraging participation. The important point to note is that attackers of the network would also be incentivized to lie, therefore, the training process needs to be trustless and verifiable.
- **Privacy Preservation:** Ensuring data remains on local devices while still contributing to the global model.

By combining PoUW with federated learning, edge devices can collaboratively train AI models while maintaining data privacy and security.

Data Processing and Analysis Edge devices can process local data (e.g., sensor readings, user interactions) and perform analyses that contribute to broader datasets or models.

- **Real-Time Insights:** Devices analyze data in real-time, reducing latency compared to cloud-based processing.

- **Network Security:** The processing contributes to the PoUW consensus, enhancing network security.
- **Economic Incentives:** Devices are rewarded for their contributions, offsetting operational costs.

4.4.2 Zero-Knowledge Proofs (ZKPs) for Privacy-Preserving AI

Overview of Zero-Knowledge Proofs

As the proliferation of edge devices continues to surge—with smartphones, IoT sensors, and embedded systems becoming ubiquitous—the need for robust privacy-preserving mechanisms in edge AI becomes paramount. Zero-Knowledge Proofs (ZKPs) emerge as a powerful cryptographic tool that allows one party (the prover) to prove to another (the verifier) that a certain statement is true without revealing any additional information beyond the validity of the statement itself (Goldreich, 2009).

Fundamental Properties A Zero-Knowledge Proof satisfies three fundamental properties (Blum et al., 1988):

- **Completeness:** If the statement is true, an honest verifier will be convinced by an honest prover.
- **Soundness:** If the statement is false, no dishonest prover can convince the honest verifier that it is true, except with some small probability.
- **Zero-Knowledge:** If the statement is true, the verifier learns nothing other than the fact that the statement is true.

These properties ensure that the proof is both convincing and does not leak any additional information, making ZKPs ideal for privacy-preserving applications.

Types of ZKPs The three most common types of ZKPs are:

- **Interactive Zero-Knowledge Proofs:** Require back-and-forth communication between the prover and verifier (Feige et al., 1988).
- **Non-Interactive Zero-Knowledge Proofs (NIZKs):** Do not require interaction; the proof can be sent as a single message (Sahai & Waters, 2014).
- **zk-SNARKs (Zero-Knowledge Succinct Non-Interactive Argument of Knowledge):** Provide succinct proofs that are quick to verify and do not require interaction (Ben-Sasson et al., 2014). zk-SNARKs are particularly useful in blockchain applications due to their efficiency.

Applications of ZKPs in Edge AI

Privacy-Preserving Computations Edge devices can perform computations on sensitive data locally and generate ZKPs to prove the correctness of these computations without revealing the underlying data (Meiklejohn & Mercer, 2018). For example, an edge device processes biometric data to authenticate a user and generates a proof that authentication was successful without transmitting the biometric data itself. The goal state of the technology would be using fully homomorphic encryption for performing computations in a privacy-preserving manner; the leaders in this industry are Zama. However, to be able to use FHE for LLMs, we require a lot more engineering efforts on hardware acceleration.

Verifiable Federated Learning In federated learning, multiple devices collaboratively train a global model. ZKPs can ensure that each participant has correctly updated the model using their local data without revealing that data (Bonawitz et al., 2017b). For example, smartphones contribute to training a predictive text model while ensuring user typing data remains private. ZKPs provide a way to verify the integrity of the model updates without compromising privacy.

Challenges of ZKPs in Edge AI

The important aspect to note here is that currently most models are too large to fit into zero-knowledge circuits; however, using primitives offered by Giza and ezkl — what is currently possible is to prove the workflow, i.e, a given device processed the data and submitted a result.

Advantages of ZKPs in Edge AI

- **Increased Trust:**
  - **Verification Without Disclosure:** Parties can trust the results of computations without needing access to the underlying data.
  - **Tamper-Proof Proofs:** Cryptographic assurances prevent malicious alteration of proofs (Miers et al., 2013).
- **Scalability and Efficiency:**
  - **Reduced Data Transmission:** Only proofs are sent over the network, reducing bandwidth usage.
  - **Edge Processing:** Computations occur locally, reducing latency and reliance on centralized servers (Bünz et al., 2018)

Use Cases of ZKPs in Edge AI

- **Healthcare:** Wearable devices monitor vital signs and detect anomalies. These devices can generate proofs that certain health thresholds have been exceeded without revealing raw data. This protects patient privacy while allowing healthcare providers to respond to critical conditions (Y. Gao et al., 2019).
- **Financial Services:** Mobile banking apps perform fraud detection algorithms locally. These applications can prove transactions meet compliance rules without revealing transaction details (Qin et al., 2020).

– **Smart Cities and IoT:** Smart meters analyze household energy usage. These meters can prove they have correctly calculated billing information without sharing detailed consumption data (F. Li et al., 2010).

4.5. Decentralized Finance (DeFi) Models for Edge AI Resource Allocation

The convergence of Decentralized Finance (DeFi) and Edge Artificial Intelligence (Edge AI) offers a transformative approach to resource allocation in distributed computing environments. By adapting DeFi concepts such as staking, lending, and liquidity pools, Edge AI networks can create self-sustaining ecosystems where computational resources, data storage, and AI services are efficiently allocated based on supply and demand dynamics. This integration not only incentivizes resource sharing among device owners but also enhances the scalability, reliability, and performance of Edge AI applications (X. Xu et al., 2020).

4.5.1 Introduction to DeFi Concepts

Decentralized Finance (DeFi) refers to a financial system built on blockchain technology that operates without intermediaries like banks or traditional financial institutions. DeFi utilizes smart contracts on decentralized platforms to provide financial services such as lending, borrowing, and trading (Schär, 2021). The key components of DeFi relevant to Edge AI include:

Staking

Staking involves locking up tokens to support network operations, such as validating transactions, in exchange for rewards. It incentivizes participants to contribute resources and maintain network security. In Edge AI, staking can

encourage device owners to offer computational resources to the network (Y. Liu & Zhang, 2019).

Lending and Borrowing

DeFi platforms allow users to lend their assets to others and earn interest, while borrowers pay interest to access these assets. This facilitates liquidity and efficient capital utilization within the network. In the context of Edge AI, devices with surplus computational power can lend resources to those requiring additional capacity (Y. Chen et al., 2020).

Liquidity Pools

Users pool their assets into a smart contract to provide liquidity for decentralized exchanges (DEXs) and earn fees. This ensures sufficient liquidity for trading and resource exchange. For Edge AI, liquidity pools can aggregate computational resources, making them readily available for tasks as needed (S. Wang et al., 2020).

4.5.2 Applying DeFi to Edge AI

Computational resources are often limited in Edge AI due to the constraints of devices like smartphones, IoT sensors, and embedded systems. By leveraging DeFi models, these devices can participate in a decentralized marketplace for computational resources (T. Li et al., 2021).

Staking Computational Resources

Device owners can stake tokens to offer their computational resources or data storage to the network. This process works as follows:

- **Resource Providers Stake Tokens:** Indicating their commitment to provide reliable resources.

- **Earning Incentives:** Providers earn rewards proportional to the resources contributed and the duration of staking.
- **Ensuring Reliability:** If providers fail to deliver or compromise resources, their staked tokens can be slashed as a penalty (X. Zhao & Sun, 2020) or their rewards will not be distributed.

There's a lot of interesting work that has been done in the Filecoin ecosystem around this area. For example, Filmine acts as an infrastructure layer on Filecoin, providing compute and storage networks a shared layer of hardware resources to run workloads while retaining the ability to connect SPs with token holders through a liquid staking protocol.

Lending and Borrowing Computational Power

Devices with surplus resources can lend them to those in need:

- **Lenders Offer Resources:** Earning interest or fees for providing computational power or storage.
- **Borrowers Access Resources:** Paying fees to utilize additional computational power, facilitating intensive AI tasks.
- **Smart Contracts Automate Agreements:** Setting terms like duration and interest rates for lending and borrowing (H. Kim & Kim, 2021).

We foresee that this idea will propagate and be observed in decentralized inference networks, i.e, when the edge device doesn't have the required compute to run the model and get results, it will pay a device that is close to it to run the model verifiably and fetch the results.

Liquidity Pools for Resources

Creating resource liquidity pools can enhance availability:

- **Contribution to Common Pool:** Device owners contribute resources to a shared pool.
- **Dynamic Allocation:** Resources are allocated to tasks based on demand, managed by smart contracts.
- **Earning Fees:** Contributors earn a share of the fees generated from resource utilization (Singh & Chatterjee, 2020).

Token Bonding Curves for Dynamic Pricing

Token bonding curves can be utilized as a mechanism to dynamically price computational resources, data access, or AI models. This allows for real-time adjustments based on supply and demand, ensuring efficient and fair pricing within the network (L. Zhang & Wu, 2021).

4.5.3 Benefits and Challenges

Efficient Resource Utilization

DeFi models help maximize the use of idle computational power, enhancing network capacity. By efficiently allocating resources based on demand, the overall performance of Edge AI applications is improved (F. Gao & Zhou, 2020).

Economic Incentives and Decentralization

Providing financial rewards for participants encourages resource sharing. DeFi models reduce reliance on centralized servers, improving security and fault tolerance. They also lower barriers to entry, enabling widespread participation in the network (Nguyen et al., 2020).

Challenges and Solutions

- **Ensuring Honest Behavior:** Implementing reputation systems and penalties (e.g., slashing staked tokens) for malicious actors helps maintain net-

work integrity (Z. Liu & Li, 2019).

- **Variability in Device Capabilities:** Standardizing protocols and using adaptive algorithms to match tasks with appropriate devices can address differences in capabilities (M. Chen, Tworek, Jun, Yuan, Pinto, Kaplan et al., 2021).
- **Timely Resource Availability:** Employing edge caching and redundancy improves reliability and ensures resources are available when needed (Q. Wang & Duan, 2020).
- **Legal Issues Related to Data Privacy and Security:** Embedding compliance protocols in smart contracts and using encryption can help navigate regulatory concerns (V. Patel & Shah, 2021).

4.5.4 Use Cases and Examples

Collaborative AI Model Training

Multiple devices collaborate to train a shared AI model. Devices stake tokens to participate, with rewards based on their contribution. This incentivizes participation and ensures devices are committed to the training process (X. Li & Wang, 2020).

Decentralized Data Storage

Edge devices offer storage space required by AI applications. Devices lend storage to the network in exchange for tokens, while borrowers pay fees to access this storage. This expands storage capacity and reduces costs compared to centralized solutions (K. Fan et al., 2019).

Real-Time Data Processing

IoT sensors generate data that requires immediate processing. Devices can borrow computational power to process data in real-time, paying fees to resource

providers. This enables timely insights crucial for applications such as autonomous vehicles (M. Chen et al., 2018).

4.6. Federated Learning and Blockchain Integration

4.6.1 Federated Learning on the Blockchain

Federated Learning enables multiple devices or nodes to train a global model collaboratively while keeping the training data localized (Konečný et al., 2016a). Each device trains the model on its local data and sends only the model updates (e.g., gradients or weights) to a central server or aggregator (Y. Li et al., 2020). This approach reduces the risk of data breaches and complies with data protection regulations by ensuring that sensitive data never leaves the local devices.

Integrating Blockchain with Federated Learning

Ensuring Data Integrity and Traceability By recording model updates on the blockchain, federated learning systems can ensure the integrity and traceability of model parameters (Y. Lu, 2019). Each update is timestamped and linked to the contributing device, providing a transparent audit trail (C. Zhang & Zhu, 2021). This ensures that all contributions to the model are accounted for and have not been tampered with, enhancing trust among participants.

Implementing Token-Based Incentive Schemes Blockchain enables the implementation of token-based incentive schemes. Participants receive rewards in cryptocurrency or tokens proportional to their contribution to the model training, encouraging active participation and honest reporting (M. Kim et al., 2019). This economic incentive aligns the interests of individual devices with the overall

performance of the global model, fostering a more robust and accurate AI system (J. Kang et al., 2018).

Smart Contracts for Automation Smart contracts are self-executing contracts with the terms of the agreement directly written into code (Szabo, 1997). In the context of federated learning, smart contracts can automate:

- **Aggregation of Model Updates:** Collecting and combining updates from participants without manual intervention (Dai et al., 2019).
- **Verification of Contributions:** Ensuring that updates meet certain quality standards before integration (J. Kang et al., 2020).
- **Distribution of Incentives:** Automatically rewarding participants based on predefined criteria (Hua et al., 2020).

4.6.2 Benefits of Blockchain-Integrated Federated Learning

Enhanced Security and Trust

Blockchain's immutable ledger ensures that model updates are securely recorded and tamper-proof (Huang & Li, 2020). This transparency builds trust among participants who may not fully trust each other, as all transactions and contributions are verifiable by the network.

Data Privacy Compliance

By keeping raw data on local devices and only sharing model updates, the system adheres to data protection regulations like GDPR and HIPAA (Voigt & Von dem Bussche, 2017). This approach minimizes the risk of sensitive data exposure and ensures compliance with legal requirements for data privacy.

Incentive Alignment

Token rewards encourage devices to participate and contribute valuable updates, enhancing the overall performance of the global model (Y. Zhan et al., 2020). This mechanism motivates participants to invest computational resources and share high-quality data, leading to better AI models.

Decentralization and Accountability

Eliminating reliance on a central aggregator reduces the risk of single points of failure and bottlenecks. Each contribution is recorded on the blockchain, allowing for auditing and accountability (Shayan et al., 2020). Participants can verify the provenance of model updates, enhancing the integrity of the federated learning process.

4.6.3 Use Cases and Examples

FLock (Federated Learning on the Blockchain)

FLock is a platform that implements federated learning integrated with blockchain technology ('Federated Learning on Blockchain', n.d.). It leverages blockchain to securely manage model updates and incentivize participants. FLock employs smart contracts to automate the aggregation process and ensures that contributors are rewarded fairly, enhancing the robustness and scalability of federated learning systems.

Healthcare Data Collaboration

Multiple hospitals collaborate to train an AI model for disease diagnosis without sharing patient data ('Federated Learning on Blockchain', n.d.). Each hospital trains the model on its local data and shares the model updates via blockchain,

ensuring data privacy and secure collaboration. This approach accelerates medical research and improves diagnostic tools while complying with strict healthcare regulations (N. Rieke et al., 2020).

Smart Manufacturing

Factories equipped with IoT devices collaborate to optimize production processes (N. Rieke et al., 2020). Devices train local models on operational data—such as equipment performance and energy consumption—and share updates via blockchain. This improves efficiency, reduces downtime, and maintains data confidentiality among competitive manufacturers (Q. Yang, Liu, Cheng et al., 2019).

Financial Fraud Detection

Banks collaborate to detect fraudulent transactions without exposing sensitive customer data (Q. Yang, Liu, Cheng et al., 2019). Federated learning on the blockchain allows them to share insights securely and comply with regulatory requirements. This collective approach enhances fraud detection capabilities while safeguarding customer privacy (N. Rieke et al., 2020).

Chapter 5

Core Frameworks for Edge AI

5.1. Introduction

5.1.1 Purpose of the Chapter

This chapter aims to provide a comprehensive examination of the core frameworks and algorithms that enable Edge AI. The primary purpose is to delve deeply into the types of algorithms suitable for edge deployment, explore the challenges involved, and discuss the optimization techniques that make efficient edge computing possible.

5.1.2 Significance of Algorithms in Edge AI

Algorithms serve as the backbone of AI systems, determining how data is processed, analyzed, and acted upon. In the context of Edge AI, the significance of algorithms is amplified due to the unique constraints of edge devices. Unlike centralized cloud servers that boast abundant computational resources, edge devices—including smartphones, IoT sensors, and embedded systems—are limited by factors such as processing power, memory capacity, and energy availability (Shi et al., 2016e). Therefore, selecting and optimizing algorithms that can operate efficiently within these limitations is crucial for the successful deployment

of AI at the edge.

Efficient algorithms enable edge devices to process data locally, which offers several key benefits:

- **Real-Time Processing:** Optimized algorithms allow for immediate data analysis, facilitating applications that require instant responses, such as autonomous vehicles navigating dynamic environments or medical devices monitoring vital signs (Y. Kang, Hauswald, Rovinski et al., 2017).
- **Enhanced Privacy and Security:** By keeping data processing on-device, sensitive information is less exposed to potential breaches during transmission over networks. This local processing enhances user privacy and complies with data protection regulations (Union, 2016a).
- **Reduced Latency and Bandwidth Usage:** Local computation minimizes the need for data to travel to and from cloud servers, reducing latency and conserving network bandwidth. This is particularly important in scenarios where network connectivity is unreliable or bandwidth is at a premium (Mach & Becvar, 2017c; Shi et al., 2016f).

The development and implementation of appropriate algorithms are thus critical to unlocking the full potential of Edge AI.

5.1.3 Overview of Topics Covered

This chapter covers a wide range of topics related to Edge AI algorithms:

- Section 2 explores the types of algorithms suitable for edge devices, including traditional machine learning methods, deep learning models, and lightweight, efficient architectures.
- Section 3 examines the challenges in deploying algorithms on edge devices, such as computational and memory constraints, energy efficiency, real-time processing requirements, and security and privacy considerations.

- Section 4 discusses optimization techniques for Edge AI algorithms, including model compression, architecture optimization, and data optimization strategies.
- Section 5 delves into the deployment of Large Language Models (LLMs) on edge devices, presenting case studies and applications.
- Section 6 provides an overview of frameworks and tools for Edge AI development, such as TensorFlow Lite, PyTorch Mobile, and Apache TVM.
- Section 7 explores Edge AI hardware platforms and their algorithm support, including microcontrollers, single-board computers, specialized AI accelerators, and hardware-algorithm co-design approaches.
- Section 8 presents application domains and use cases for Edge AI algorithms, spanning computer vision, audio and speech processing, natural language processing, anomaly detection, and healthcare.
- Section 9 discusses federated learning and collaborative Edge AI, highlighting privacy-preserving techniques and real-world implementations.
- Section 10 addresses security and privacy aspects of Edge AI algorithms, including threat models, adversarial attacks, and defense mechanisms.
- Section 11 looks towards the future, exploring next-generation Edge AI algorithms, advances in hardware, integration with emerging technologies, and open research challenges.
- Finally, Section 12 concludes the chapter with a summary of key points and final thoughts on the road ahead for Edge AI algorithms.

5.2. Types of Algorithms Suitable for Edge Devices

5.2.1 Traditional Machine Learning Algorithms

Traditional machine learning algorithms are often less computationally intensive compared to deep learning models, making them suitable for edge deployment

in certain scenarios.

Decision Trees and Random Forests Decision Trees are hierarchical models that make decisions based on feature values, splitting data into branches to reach a prediction (Breiman et al., 1984). They are simple to implement and require minimal computational resources, which makes them suitable for edge devices handling low-dimensional data.

Random Forests are ensemble methods that combine multiple decision trees to improve predictive accuracy and control over-fitting (Breiman, 2001). While more computationally demanding than a single decision tree, random forests can still be feasible on edge devices, especially when the number of trees is limited. These algorithms can be applied in environmental monitoring using sensor data (Gama et al., 2014) and anomaly detection in IoT networks (Bhattacharya & Pal, 2015).

Support Vector Machines (SVMs) Support Vector Machines (SVMs) are supervised learning models used for classification and regression tasks (Cortes & Vapnik, 1995). They find the optimal hyperplane that separates data into classes. With kernel tricks, SVMs can handle non-linear data, but this increases computational complexity.

To deploy SVMs on edge devices, it is better to use linear SVMs for lower computational overhead (Joachims, 2006). Methods like reduced-set vectors can decrease memory usage (Burges, 1996).

K-Nearest Neighbors (KNNs) K-Nearest Neighbors (KNN) is a non-parametric method used for classification and regression by analyzing the k closest training examples in the feature space (Cover & Hart, 1967). This algorithm is simple to understand and implement. However, it requires computation over the entire dataset for each prediction and needs to store all the training data in memory, which can be impractical. To deploy KNNs onto edge devices, limit dataset size

or use dimensionality reduction techniques (Jolliffe, 2002), or implement efficient data structures like KD-trees for faster nearest neighbor searches (Bentley, 1975).

5.2.2 Deep Learning Algorithms

Deep Learning has fundamentally changed the way AI is used in our lives, but these algorithms are typically resource-intensive. However, certain architectures and techniques can make them suitable for edge deployment.

Convolutional Neural Networks (CNNs) Convolutional Neural Networks (CNNs) are specialized neural networks designed for processing grid-like data structures such as images (LeCun et al., 1998). The convolutional layers share weights, reducing the number of parameters compared to fully connected networks. For edge devices, smaller architectures (e.g., AlexNet, VGGNet) (Krizhevsky et al., 2012) are advised, along with model compression techniques and lightweight CNN variants.

Recurrent Neural Networks (RNNs) and Long Short-Term Memory Networks (LSTMs) Recurrent Neural Networks (RNNs) are designed for sequential data, while Long Short-Term Memory (LSTM) networks address the vanishing gradient problem in RNNs (Cho et al., 2014; Hochreiter & Schmidhuber, 1997). These are effective for tasks involving time-series data and natural language processing. For edge deployment, simpler architectures like Gated Recurrent Units (GRUs) are preferred, and sequence lengths can be limited to reduce computational demands.

Graph Neural Networks (GNNs) Graph Neural Networks (GNNs) operate on graph-structured data, capturing relationships between nodes (Z. Wu et al., 2021). GNNs are useful in applications such as social network analysis and molecular property prediction. For edge deployment, it is advisable to simplify

models by reducing layers or using sparse matrix operations (J. Chen et al., 2018).

5.2.3 Lightweight and Efficient Models

Researchers have developed models specifically designed to be lightweight and computationally efficient for edge devices without significantly compromising performance.

MobileNets MobileNets are a class of efficient models designed for mobile and embedded vision applications (A. G. Howard et al., 2017a). They use depthwise separable convolutions to reduce computation and model size. Applications include real-time object detection on smartphones (Sandler et al., 2018) and image classification in resource-constrained environments.

SqueezeNet SqueezeNet aims to achieve AlexNet-level accuracy with 50x fewer parameters (Iandola et al., 2016). It introduces the Fire module, which squeezes and expands channels to reduce parameters. Applications include deployment in IoT devices with limited memory (J. Wu et al., 2016).

EfficientNet EfficientNet proposes a family of models that scale up networks in a balanced way using a compound coefficient (M. Tan & Le, 2019). It scales network width, depth, and resolution. Smaller variants like EfficientNet-B0 (M. Tan & Le, 2021) are ideal for edge deployment, and further optimizations can be applied via quantization and pruning.

5.3. Challenges in Deploying Algorithms on Edge Devices

Deploying AI algorithms on edge devices presents a unique set of challenges stemming from the inherent limitations of these devices. Unlike cloud servers, edge devices such as smartphones, IoT sensors, and embedded systems have constrained computational resources, limited memory, and strict energy budgets. This section discusses the primary challenges faced when deploying algorithms on edge devices and the implications for Edge AI development.

5.3.1 Computational and Memory Constraints

Computational Limitations Edge devices are equipped with processors that are significantly less powerful than those found in cloud servers. They often lack specialized hardware like high-end GPUs or TPUs that accelerate complex computations (Y. Chen et al., 2019). This limitation affects the feasibility of deploying computationally intensive algorithms, particularly deep learning models with millions or billions of parameters.

- **Processing Power:** Limited CPU/GPU capabilities lead to longer processing times for complex models, making real-time inference challenging (H. Li et al., 2019).
- **Parallelism:** Edge devices may not support the level of parallel computation required by certain algorithms, hindering performance (J. Ren et al., 2020).

Memory Constraints Memory is another critical resource that is limited on edge devices. RAM and storage capacities are often insufficient to accommodate large models and datasets.

- **RAM Limitations:** Running large models can exceed the available RAM, causing failures or the need for swapping, which is not feasible in many embedded systems (K. Zhang et al., 2016a).
- **Storage Space:** Persistent storage constraints limit the ability to store large models or datasets locally (Abolfazli et al., 2014).

Implications for Algorithm Development

- **Model Size Reduction:** These challenges necessitate the use of model compression techniques.
- **Algorithm Selection:** It is favored to utilize algorithms with lower computational complexity and smaller memory footprints.

5.3.2 Energy Efficiency and Power Consumption

Edge devices often operate on limited power sources, such as batteries, making energy efficiency a paramount concern.

Power Constraints

- **Battery Life:** Prolonged computational activities drain battery life, reducing the usability of mobile and portable devices (Lane et al., 2016).
- **Thermal Considerations:** Intensive computations generate heat, which can affect device performance and longevity (Rudenko et al., 1998).

Energy Consumption of Algorithms

- **Complex Models:** Deep neural networks require substantial energy for both training and inference (M. Wang et al., 2022).
- **Continuous Operation:** Always-on applications (e.g., voice assistants) necessitate algorithms that are energy-efficient to prevent rapid battery depletion (Y. Lin et al., 2018).

Strategies to Mitigate Energy Consumption

- **Algorithm Optimization:** Designing algorithms that require fewer computations.
- **Hardware Acceleration:** Utilizing specialized low-power hardware accelerators (e.g., NPUs) (Moons & Verhelst, 2016).
- **Duty Cycling:** Turning off or scaling down computations when not needed (Mishra et al., 2008).

5.3.3 Real-Time Processing Requirements

Many edge applications demand real-time or near-real-time processing to be effective.

Latency Sensitivity

- **Immediate Response Needed:** Applications like autonomous driving, industrial automation, and health monitoring require instant decision-making (Sallouha et al., 2017).
- **User Experience:** High latency can degrade the user experience in applications like augmented reality or interactive assistants (Azuma, 1997).

Challenges in Achieving Real-Time Performance

- **Processing Delays:** Limited computational resources can lead to slower processing times (Satyanarayanan, 2017d).
- **Data Throughput:** Handling high-frequency data streams can overwhelm the device's processing capabilities (Hung et al., 2016).

Approaches to Meet Real-Time Requirements

- **Algorithm Simplification:** Using models with fewer layers or parameters to reduce inference time (J. Chen et al., 2016a).

- **Asynchronous Processing:** Implementing algorithms that can process data in an event-driven manner (Premsankar et al., 2018a).
- **Prioritization:** Focusing computational resources on critical tasks while deferring less important ones.

5.3.4 Security and Privacy Considerations

Deploying AI algorithms on edge devices introduces unique security and privacy challenges.

Data Privacy

- **Sensitive Information:** Edge devices often handle personal or sensitive data (e.g., health metrics, location data) (Nakamoto, 2008).
- **Local Data Processing Risks:** While local processing enhances privacy, it also places the burden of data protection on devices that may not be secure (L. Zhang et al., 2016).

Security Threats

- **Physical Access:** Edge devices may be more susceptible to physical tampering or theft (Chan & Yeung, 2012).
- **Vulnerabilities in Software:** Limited computational resources may prevent the use of robust security protocols, making devices vulnerable to attacks (J. Tang et al., 2016).

Algorithmic Risks

- **Adversarial Attacks:** Algorithms can be fooled by carefully crafted inputs, leading to incorrect outputs (Goodfellow et al., 2015a).
- **Model Extraction:** Attackers may attempt to reverse-engineer models to steal intellectual property or find vulnerabilities (Fredrikson et al., 2015).

Mitigation Strategies

- **Encryption:** Employing data encryption for stored and transmitted data (Gentry, 2009).
- **Secure Bootloaders and Firmware:** Ensuring only authenticated software runs on the device (Koeberl et al., 2014).
- **Privacy-Preserving Techniques:** Utilizing federated learning and differential privacy to protect user data (B. McMahan et al., 2017).
- **Regular Updates:** Implementing mechanisms for over-the-air updates to patch security vulnerabilities (Ammar et al., 2018).

Balancing Performance and Security

- **Resource Allocation:** Security measures consume computational resources, potentially impacting performance (Rawat et al., 2015).
- **Design Trade-offs:** Developers must balance the need for security with the constraints of edge devices.

5.4. Optimization Techniques for Edge AI Algorithms

Optimizing AI algorithms for edge deployment is essential due to the limited computational resources, memory, and energy constraints of edge devices. This section explores various optimization techniques that enable efficient execution of AI models on edge hardware without significantly compromising performance.

5.4.1 Model Compression

Model compression aims to reduce the size and computational complexity of AI models, making them more suitable for deployment on resource-constrained devices.

Quantization Techniques Quantization reduces the precision of the numerical values (weights and activations) in neural networks, typically from 32-bit floating-point to lower-bit representations like 16-bit, 8-bit, or even binary (Han et al., 2016c). Quantization techniques include:

- **Uniform Quantization:** Applies a consistent scale across all weights and activations (Rastegari et al., 2016).
- **Dynamic Range Quantization:** Quantizes weights to 8-bit integers while leaving activations in floating-point, reducing model size with minimal impact on latency (TensorFlow, 2021c).
- **Quantization-Aware Training (QAT):** Simulates quantization effects during training to preserve accuracy in the lower-precision model (Z. Zhao et al., 2019).
- **Post-Training Quantization (PTQ):** Applies quantization to a pre-trained model without retraining, offering a quick optimization at the potential cost of accuracy (Nagel et al., 2020).

Pruning and Weight Sharing Pruning eliminates redundant or less significant weights from a neural network to reduce its complexity (Han, Pool et al., 2015).

- **Unstructured Pruning:** Removes individual weights based on a threshold, leading to sparse weight matrices (LeCun et al., 1990).
- **Structured Pruning:** Removes entire neurons, filters, or channels, resulting in smaller models that are more efficient on standard hardware (H. Li et al., 2017).
- **Weight Sharing:** Forces multiple weights in a neural network to share the same value (W. Chen et al., 2015).
- **HashNet:** Utilizes hash functions to group weights into bins, sharing the same value within each bin (Leng et al., 2018).

- **Tensor Factorization:** Decomposes large weight tensors into smaller components for parameter sharing (Novikov et al., 2015).

Pruning and weight sharing can reduce model size and computational requirements by up to 90% with minimal loss in accuracy (He et al., 2017).

Knowledge Distillation Knowledge Distillation transfers knowledge from a large, complex model (teacher) to a smaller, efficient model (student) (Hinton et al., 2015b).

- **Soft Targets:** The student model learns from the teacher's output probabilities, capturing more information than hard labels (Buciluǎ et al., 2006).
- **Loss Function:** Combines standard loss with a distillation loss that measures the difference between teacher and student outputs (Ba & Caruana, 2014).

This technique enables the student model to mimic the teacher's performance while being significantly smaller and faster, making it ideal for edge deployment (Romero et al., 2015).

Low-Rank Factorization Low-Rank Factorization approximates weight matrices with lower-rank representations to reduce the number of parameters (Sainath et al., 2013).

- **Singular Value Decomposition (SVD):** Decomposes weight matrices into products of smaller matrices (Denton et al., 2014).
- **Tensor Decomposition:** Extends matrix factorization to multi-dimensional tensors used in convolutional layers (Phan et al., 2016).

By reducing redundancy in weight matrices, low-rank factorization decreases computational load and memory usage with minimal impact on model accuracy (Y. Kim et al., 2016).

5.4.2 Architecture Optimization

Optimizing the neural network architecture itself can lead to significant efficiency gains on edge devices.

Neural Architecture Search (NAS) Neural Architecture Search automates the design of neural network architectures optimized for specific tasks and constraints (Zoph & Le, 2017).

- **Search Methods:** Includes reinforcement learning, evolutionary algorithms, and gradient-based optimization (H. Liu et al., 2019).
- **Hardware-Aware NAS:** Considers hardware constraints like latency, memory, and power consumption during the search process (Cai et al., 2019b).
- **Examples:**
  - **MnasNet:** Balances accuracy and latency for mobile devices using a multi-objective NAS approach (M. Tan et al., 2019).
  - **FBNet:** Employs differentiable NAS to generate efficient architectures for edge devices (B. Wu et al., 2019).

Hardware-Aware Model Design Designing models with specific hardware characteristics in mind enhances efficiency (Sze, Chen et al., 2017).

- **Specialized Layers:** Utilizing operations optimized for the target hardware, such as depth-wise separable convolutions (Sandler et al., 2018).
- **Latency and Memory Constraints:** Incorporating these constraints into the design process to ensure models meet real-time requirements (A. Howard et al., 2019b).
- **Energy Efficiency:** Optimizing for lower power consumption without sacrificing performance (Dong et al., 2019).

TinyML Approaches TinyML focuses on implementing machine learning models on microcontrollers and ultra-low-power devices (Warden & Situnayake, 2019).

- **Model Optimization:** Aggressively reduces model size and complexity to fit within kilobytes of memory (David et al., 2021).
- **Efficient Inference Engines:** Uses lightweight inference engines optimized for microcontrollers (Pouchet, Singh et al., 2017).
- **Applications:** Includes keyword spotting, gesture recognition, and simple anomaly detection (**sze2017hardware**).

TinyML expands the reach of AI by enabling machine learning capabilities on the smallest and most resource-constrained devices (Strommer et al., 2020).

5.4.3 Data Optimization

Optimizing data used for training and inference can improve model efficiency and performance on edge devices.

Data Augmentation Strategies Data Augmentation increases the diversity of training data without collecting new samples (Taylor & Nitschke, 2018).

- **Image Transformations:** Includes rotations, flips, crops, and color adjustments (Krizhevsky et al., 2012).
- **Synthetic Data Generation:** Uses generative models to create new data samples (Wei & Zou, 2019).
- **Domain-Specific Augmentation:** Tailors augmentation techniques to the specific characteristics of the deployment environment (Shorten & Khoshgoftaar, 2019).

Dataset Reduction Techniques Reducing the size of datasets can make on-device training and inference more feasible (Paulin, Seldin et al., 2014).

- **Data Pruning:** Removes redundant or less informative data points (Bachem, Lucic et al., 2017).
- **Core-Set Selection:** Identifies a small, representative subset of the data that maintains model performance (Creswell et al., 2018).
- **Compression Techniques:** Applies methods like quantization to data to reduce storage requirements (Bengio et al., 2013).

Dataset reduction helps manage limited storage and memory resources on edge devices, enabling efficient data handling.

5.5. Deployment of Large Language Models (LLMs) on Edge Devices

5.5.1 Introduction to LLMs in Edge AI

The deployment of Large Language Models (LLMs) on edge devices has enabled a new generation of applications that leverage advanced natural language processing capabilities directly on devices like smartphones, tablets, and IoT gadgets. This section delves into specific case studies, exploring the techniques used to overcome challenges and the benefits achieved.

- **On-Device Conversational Agents:** On-device conversational agents have become increasingly sophisticated, offering personalized and responsive user experiences without relying heavily on cloud services. Deploying LLMs on edge devices enhances privacy, reduces latency, and allows for offline functionality.

  Case Study: Apple Siri's On-Device Processing (Apple Intelligence): Apple has integrated on-device processing for Siri, enabling the assistant to handle requests without internet connectivity for certain tasks

(Inc., 2021a). **Technical Implementation:** Apple's Neural Engine accelerates machine learning computations, allowing for efficient execution of LLMs on devices (Sandler et al., 2018). Techniques like quantization and pruning reduce the model size while maintaining performance (Han et al., 2016b). All voice processing is done locally, ensuring user data remains on the device (Narayanan et al., 2019). **Benefits:** Immediate response times enhance user experience (Lane & Warden, 2018). Local processing prevents sensitive data from being sent to servers (Inc., 2021b).

– **Case Study: Google Assistant's Edge AI:** Google Assistant has incorporated on-device speech recognition and natural language understanding (He et al., 2019b). **Technical Implementation:** It uses Recurrent Neural Network Transducer (RNN-T) models optimized for edge devices (Y. Zhang et al., 2020) and employs knowledge distillation and quantization techniques to compress models for efficient on-device deployment (Kwon et al., 2020). **Benefits:** This approach allows users to access certain features without internet connectivity (Blog, 2020), and optimized models consume less power, extending device battery life (Kumar et al., 2017).

  The challenges with building on-device conversational agents lie in balancing model complexity with device constraints. A solution lies in implementing adaptive computation and leveraging specialized hardware accelerators (W. Jiang et al., 2020a).

– **Real-Time Translation Services:** On-device real-time translation enables users to communicate across languages instantly, even without internet access. Deploying LLMs for translation tasks on edge devices enhances privacy and reliability.

  Case Study: Google Translate Offline Mode: Google Translate offers offline language packs that allow translation without internet connectivity (Google, 2021). **Technical Implementation:** It uses LLMs optimized for mobile devices through quantization and pruning (Y. Kim et al., 2020),

which reduces model size by converting weights to lower-precision formats (M. Wu et al., 2020). **Benefits:** This facilitates communication in areas with limited connectivity (Heafield et al., 2016) and keeps user data on-device, safeguarding sensitive information (Microsoft, 2021).

Case Study: Microsoft Translator: Microsoft Translator provides offline translation capabilities through downloadable language packs (Wen et al., 2015). **Technical Implementation:** It employs compressed LSTM-based models suitable for edge devices (Sutskever et al., 2014) and optimizes inference speed while reducing memory footprint (Kudo & Richardson, 2018). **Benefits:** This also facilitates communication in areas with limited connectivity (Z. Tang et al., 2020) and keeps user data on-device, safeguarding sensitive information (Sun et al., 2020).

Applying advanced compression techniques like knowledge distillation to retain model effectiveness (Sanh et al., 2019b) can ensure translation accuracy.

– **Privacy-Preserving Text Processing:** Processing sensitive text data on-device is crucial for applications in healthcare, finance, and personal communications. Deploying LLMs on edge devices enables privacy-preserving text analysis.

Case Study: On-Device Health Data Analysis: Health apps analyze user data to provide insights while complying with privacy regulations (J. Xu et al., 2020). **Technical Implementation:** Trusted Execution Environments (TEEs) and Secure Enclaves can be used to protect data during processing (Ltd., 2021). Lightweight LLMs tailored for medical text analysis (Wolf et al., 2021) can be employed. **Benefits:** This ensures regulatory compliance by meeting standards such as HIPAA, as data remains local (of Health & Human Services, 2021), enhancing confidence in data security and privacy (Shokri & Shmatikov, 2015).

Case Study: Financial Transaction Monitoring: Financial apps de-

tect fraudulent activities by analyzing transaction data on-device (R. Zhang et al., 2020). **Technical Implementation:** Models are trained across multiple devices without centralizing data (B. McMahan et al., 2016) through federated learning. Data is encrypted during processing and storage (Boneh & Lipton, 1996). **Benefits:** This means sensitive financial data is not transmitted over networks (Pérez et al., 2021) and enables real-time detection of anomalies without server dependency (**ward2020**).

This approach ensures data protection without compromising performance by utilizing model optimization with robust security protocols (Fredrikson et al., 2015).

5.5.2 Technical Innovations Underpinning the Deployment of LLMs on the Edge

- **Advanced Model Compression:** Techniques such as pruning and quantization remove redundant parameters and reduce precision to minimize model size (M. Wu et al., 2020).
  - **Pruning and Quantization:** Removes redundant parameters and reduces precision to minimize model size (Z. Yang & Yu, 2017).
  - **Knowledge Distillation:** Transfers knowledge from larger models to smaller ones without significant loss in performance (Hinton et al., 2015b).
- **Hardware Acceleration:** Specialized processors like Edge TPUs and NPUs are designed for efficient neural network inference on edge devices (Ignatov et al., 2019).
  - **Edge TPUs and NPUs:** Specialized processors designed for efficient neural network inference on edge devices (Ignatov et al., 2019).
  - **Optimized Libraries:** Software frameworks like TensorFlow Lite and PyTorch Mobile enable efficient model deployment (TensorFlow, 2024).

- **Adaptive Inference Techniques:** These techniques improve the efficiency of inference processes in LLMs on edge devices.
  - **Early Exiting Mechanisms:** Models can exit inference at intermediate layers if confidence thresholds are met (Horowitz, 2014).
  - **Dynamic Inference Paths:** Allocates resources selectively, processing simpler inputs with fewer computations (X. Lu & Li, 2020a).
- **Ongoing Research:** The field is constantly evolving, with ongoing research aimed at further optimizing LLMs for edge deployments. Notable areas of focus include:
  - **Federated Learning Enhancements:** Improving privacy-preserving training across devices (Q. Yang, Liu, Chen & Tong, 2019).
  - **Energy-efficient Architectures:** Designing models and hardware that consume less power (Horowitz, 2014).
  - **Edge-to-Edge Collaboration:** Enabling devices to share insights directly, forming decentralized intelligence networks (X. Lu & Li, 2020a).

5.6. Frameworks and Tools for Edge AI Development

Deploying AI models on edge devices requires specialized frameworks and tools that cater to the unique constraints of limited computational resources, memory, and power. This subsection provides an in-depth examination of prominent frameworks and tools that facilitate the development and deployment of Edge AI applications. We will explore their features, capabilities, and how they address the challenges inherent in edge computing.

5.6.1 Overview of Edge AI Frameworks

Edge AI frameworks are designed to bridge the gap between complex AI models and resource-constrained devices. They provide tools for model optimization, conversion, and efficient execution on various hardware platforms. Key considerations for these frameworks include:

- **Model Optimization Techniques:** Quantization, pruning, and compression to reduce model size and computational load.
- **Hardware Acceleration:** Support for leveraging specialized hardware like GPUs, NPUs, and TPUs.
- **Cross-Platform Compatibility:** Ability to deploy on multiple operating systems and hardware architectures.
- **Ease of Integration:** User-friendly APIs and tools for seamless integration into applications.

5.6.2 TensorFlow Lite

TensorFlow Lite is an open-source deep learning framework for on-device machine learning, developed by Google (TensorFlow, 2024). It is a lightweight solution for mobile and embedded devices.

Features and Capabilities

- **Lightweight Interpreter:** Designed for efficiency on devices with limited resources, the TensorFlow Lite interpreter has a small binary size and minimal runtime dependencies (Warden, 2019).
- **Optimized Kernels:** Provides a set of optimized operations (kernels) for common neural network functions, ensuring efficient execution ('Optimizing TensorFlow models for mobile and edge devices', 2024).
- **Hardware Acceleration:**

    - **GPU Delegates:** Utilizes mobile GPUs for acceleration ('GPU Delegate for TensorFlow Lite', 2024).
    - **Edge TPU Support:** Compatible with Google's Edge TPU for enhanced performance ('Coral Edge TPU: Supercharging inference at the edge', 2024).
    - **NNAPI Delegates:** Interfaces with Android's Neural Networks API (NNAPI) for hardware acceleration ('NNAPI Delegate for TensorFlow Lite', 2024).
  - **Pre-built and Custom Models:** Supports both out-of-the-box models and custom models converted from TensorFlow ('Pre-trained models for TensorFlow Lite', 2024).
  - **Cross-Platform Support:** Compatible with Android, iOS, and embedded Linux platforms.

Model Conversion and Optimization

  - **TensorFlow Lite Converter:** Converts TensorFlow models into the TensorFlow Lite format (.tflite) ('TensorFlow Lite Converter', 2024). Supports SavedModel, Keras, and concrete functions.
  - **Quantization Techniques:**
    - **Post-Training Quantization:** Reduces model size and increases inference speed by converting weights to lower precision
    - **Quantization-Aware Training:** Incorporates quantization during training to maintain higher accuracy
  - **Model Optimization Toolkit:** Provides tools for pruning and clustering to further optimize models
  - **Selective Registration:** Reduces binary size by including only necessary operations

5.6.3 PyTorch Mobile

PyTorch Mobile is a platform developed by Facebook for deploying PyTorch models on mobile and embedded devices (PyTorch, 2020).

Deployment Workflow

- **Model Exporting:**
  - **TorchScript:** Converts PyTorch models to a serialized format that can be loaded in a non-Python environment
  - **Tracing:** Records operations by running an example input through the model.
  - **Scripting:** Converts models with dynamic control flows.
- **Integration with Mobile Platforms:**
  - **Android:** Provides Java bindings and example applications
  - **iOS:** Offers Swift and Objective-C APIs for model loading and inference.
  - **Custom Mobile Build:** Enables creation of a smaller runtime by including only necessary components.

Performance Considerations

- **Quantization:** Supports both static and dynamic quantization methods to optimize models (PyTorch, 2024).
- **Optimized Backend Engines:**
  - **QNNPACK:** Optimized for quantized 8-bit operations on ARM CPUs
  - **FBGEMM:** Optimized for server-side CPUs but can be used on some edge devices.
- **Memory Management:** Provides tools to monitor and reduce memory usage during inference.

- **Selective Operator Loading:** Includes only the operators required by the model to reduce binary size.

5.6.4 ONNX and ONNX Runtime

The Open Neural Network Exchange (ONNX) is an open standard for representing machine learning models, enabling interoperability between different frameworks (ONNX, 2024) ONNX Runtime is a high-performance inference engine for ONNX models (Microsoft, 2024).

Interoperability Between Frameworks

- **Model Conversion:** Supports conversion from frameworks like TensorFlow, PyTorch, Keras, and more. Facilitates moving models between development environments and deployment platforms.
- **Standardization:** Provides a unified format to reduce friction in deploying models across different systems.
- **Tooling Support:** Offers a rich ecosystem of tools for model inspection, visualization, and optimization.

Edge Deployment Support

- **ONNX Runtime Mobile:** Tailored for mobile and embedded devices with a focus on minimal binary size.
- **Optimizations:**
  * Supports graph optimizations, operator fusion, and memory footprint reductions.
  * Quantization Tools: Facilitates model quantization to improve performance on edge devices.

- **Hardware Acceleration:** Integrates with hardware-specific libraries and accelerators, including NVIDIA TensorRT, Intel OpenVINO, and ARM Compute Library.
- **Cross-Platform Execution:** Runs on various operating systems and hardware architectures, providing flexibility in deployment.

5.6.5 Apache TVM

Apache TVM is an open-source deep learning compiler stack that enables high-performance machine learning model deployment across a variety of hardware backends (Apache TVM, 2024).

Automated Model Optimization

- **Model Compilation:** Converts high-level models into optimized low-level code tailored for specific hardware.
- **AutoTVM:** An automated tensor optimization framework that tunes kernel performance based on hardware characteristics.
- **Relay IR:** An intermediate representation that provides optimization passes and supports multiple frontends.
- **VTA (Versatile Tensor Accelerator):** An open-source deep learning accelerator compatible with TVM for research and development.

Cross-Platform Deployment

- **Hardware Abstraction:** Supports CPUs, GPUs, FPGAs, and specialized accelerators across vendors.
- **MicroTVM:** Enables deployment on microcontrollers and other devices with very limited resources.
- **Edge Device Support:** Optimizes models for edge devices by considering resource constraints during compilation.

- **Community and Extensibility:** Active development community contributing to a wide range of hardware targets and optimizations.

Other Notable Tools

- **NVIDIA TensorRT:** NVIDIA TensorRT is a high-performance deep learning inference optimizer and runtime library designed for NVIDIA GPUs (NVIDIA, 2024).
  - **Features:**
    - **Graph Optimizations:** Layer and tensor fusion, kernel auto-tuning.
    - **Precision Calibration:** Supports FP32, FP16, and INT8 precisions for performance scaling.
    - **Dynamic Tensor Memory:** Efficient memory usage during inference.
    - **Model Conversion:** Imports models from frameworks like TensorFlow, PyTorch, and ONNX.
    - **Multi-GPU Support:** Efficiently distributes inference across multiple GPUs.
- **OpenVINO:** Developed by Intel, OpenVINO optimizes deep learning models for Intel hardware, providing tools for model optimization and deployment ('OpenVINO Toolkit', 2024).
  - **Model Optimizer:** Converts models from various frameworks into an intermediate representation optimized for inference.
  - **Inference Engine:** Provides a unified API for executing models on different Intel hardware platforms.
  - **Support for Various Devices:** Includes CPUs, GPUs, VPUs, and FPGAs.

- **Edge Impulse:** Edge Impulse is a development platform focused on embedded

machine learning for edge devices, particularly microcontrollers and small CPUs (Edge Impulse, n.d.).

- **Features:**
  - **Data Acquisition:** Tools for collecting and managing sensor data.
  - **Model Training:** Automated pipeline for training models suitable for edge devices.
  - **Optimization:** EON Compiler optimizes models to run with minimal footprint.
- **Deployment:** Generates code and libraries for a variety of hardware platforms.
- **User Interface:** Provides a web-based interface for managing projects, with support for collaboration.
- **Community and Support:** Active community forums and extensive documentation.

5.7. Edge AI Hardware Platforms and Their Algorithm Support

Edge AI relies heavily on hardware platforms that can efficiently execute AI algorithms within the constraints of limited power, computational resources, and memory. This section provides an in-depth exploration of various hardware platforms suited for Edge AI applications, including microcontrollers, single-board computers, specialized AI accelerators, and the concept of hardware-algorithm co-design.

5.7.1 Microcontrollers and Microprocessors

Microcontrollers and microprocessors form the backbone of many edge devices, offering a balance between performance, power consumption, and cost. They are

essential for deploying lightweight AI models and running simple inference tasks directly on devices like sensors, wearables, and embedded systems.

ARM Cortex Series

The ARM Cortex series is a family of 32-bit and 64-bit RISC (Reduced Instruction Set Computing) microprocessors widely used in embedded systems and mobile devices (ARM Ltd., n.d.-b). They are known for their energy efficiency and performance, making them suitable for Edge AI applications.

Key Features

- **Cortex-M Series:** Designed for microcontrollers with ultra-low power consumption (Yiu, 2013). Suitable for simple AI tasks like sensor data processing and anomaly detection.
- **Cortex-A Series:** Targets higher performance applications, often used in smartphones and tablets (ARM Ltd., n.d.-a). Supports operating systems like Linux and Android, enabling more complex AI applications.
- **NEON SIMD Architecture:** Single Instruction Multiple Data (SIMD) extension for accelerating multimedia and signal processing tasks (ARM Ltd., n.d.-f). Enhances performance for parallelizable AI algorithms.

Algorithm Support

- **CMSIS-NN Library:** A collection of efficient neural network kernels optimized for Cortex-M processors (ARM Ltd., n.d.-c). Enables deployment of deep learning models on microcontrollers with limited resources.
- **Arm Compute Library:** Provides optimized functions for machine learning and computer vision on Cortex-A CPUs and Mali GPUs (ARM Ltd., n.d.-d). Supports frameworks like TensorFlow Lite Micro and PyTorch Mobile.

Use Cases

- **Wearable Devices:** Health monitoring and activity recognition using lightweight neural networks (Zhou et al., 2019).
- **IoT Sensors:** On-device data preprocessing and anomaly detection to reduce data transmission (S. Li et al., 2020).

RISC-V Architecture

RISC-V is an open-source instruction set architecture (ISA) that offers extensibility and customization, making it attractive for specialized edge computing applications (RISC-V Foundation, n.d.).

Key Features

- **Open-Source ISA:** Free and open, allowing for customization and optimization for specific applications (Waterman & Asanović, 2017).
- **Scalability:** Supports a range of implementations from small microcontrollers to high-performance processors (Celio et al., 2017).
- **Extensions for AI:** Custom extensions can be added to accelerate AI workloads (Puggelli et al., 2018).

Algorithm Support

- **AI-Optimized Cores:** Projects like SiFive's AI cores incorporate vector extensions for AI acceleration (SiFive, n.d.).
- **Software Ecosystem:** Support for machine learning libraries and frameworks is growing, with ports of TensorFlow Lite and TVM (Haj-Ali et al., 2019).

Use Cases

- **Edge Computing Devices:** Customizable processors for specific AI tasks in industrial automation and robotics (Dinechin et al., 2013).
- **Research and Development:** Academia and industry use RISC-V for exploring new hardware-software co-design approaches (Asanović & Patterson, 2014).

5.7.2 Single Board Computers

Single Board Computers (SBCs) offer more computational power than microcontrollers and are suitable for running more complex AI models. They provide a versatile platform for Edge AI development.

Raspberry Pi

The Raspberry Pi is a low-cost, credit-card-sized computer that has gained popularity for education, prototyping, and hobbyist projects (Raspberry Pi Foundation, n.d.-b).

Key Features

- **Broad Compatibility:** Runs a full Linux operating system, supporting a wide range of software and programming languages (Upton & Halfacree, 2014).
- **GPIO Pins:** General-purpose input/output pins for interfacing with sensors and other hardware (Raspberry Pi Foundation, n.d.-a).
- **Multiple Models:** Variants like Raspberry Pi 4 offer up to 8GB RAM and a quad-core CPU (Raspberry Pi Foundation, n.d.-c).

Algorithm Support

- **Machine Learning Frameworks:** Supports TensorFlow Lite, PyTorch, and OpenCV for AI applications (TensorFlow, n.d.-a).

- **Hardware Acceleration:** Limited built-in acceleration, but compatible with external accelerators like the Google Coral USB Edge TPU (Google Coral, n.d.-e).

Use Cases

- **Computer Vision:** Image and video processing for surveillance, robotics, and home automation (Rosebrock, 2019).
- **Edge Analytics:** Data processing and analytics for IoT applications (White, 2012).

NVIDIA Jetson Nano

The NVIDIA Jetson Nano is a powerful SBC designed specifically for AI and machine learning tasks at the edge (NVIDIA, n.d.-d).

Key Features

- **GPU Acceleration:** Equipped with a 128-core NVIDIA Maxwell GPU for parallel processing (NVIDIA, n.d.-f).
- **High Performance:** Capable of running complex neural networks with up to 472 GFLOPs of compute performance (Mittal, 2019).
- **Developer-Friendly:** Supports Ubuntu Linux and comes with NVIDIA's JetPack SDK (NVIDIA, n.d.-c).

Algorithm Support

- **Deep Learning Frameworks:** Optimized versions of TensorFlow, PyTorch, and Caffe are available (NVIDIA, n.d.-b).
- **CUDA and cuDNN:** NVIDIA's libraries for GPU acceleration of AI algorithms (NVIDIA, n.d.-a).

- **TensorRT:** A platform for high-performance deep learning inference optimized for NVIDIA hardware (NVIDIA, n.d.-h).

Use Cases

- **Autonomous Machines:** Robotics, drones, and autonomous vehicles that require real-time AI processing (Leake et al., 2018).
- **Smart Cameras:** Advanced image recognition and analytics for security and retail applications (Girshick et al., 2016).

5.7.3 Specialized AI Accelerators

Specialized AI accelerators are hardware designed specifically to accelerate AI workloads, offering high performance with low power consumption.

Google Edge TPU

The Google Edge TPU is a purpose-built ASIC (Application-Specific Integrated Circuit) designed to run AI at the edge (Google Coral, n.d.-d).

Key Features

- **High Efficiency:** Delivers up to 4 TOPS (Tera Operations Per Second) while consuming minimal power (Hong & Gonzalez, 2020).
- **Compatibility:** Supports TensorFlow Lite models converted to the Edge TPU format (TensorFlow, n.d.-b).
- **Form Factors:** Available as a USB accelerator, PCIe card, and integrated into development boards like the Coral Dev Board (Google Coral, n.d.-a).

Algorithm Support

- **Model Compatibility:** Supports a subset of TensorFlow operations optimized for the Edge TPU (Google Coral, n.d.-c).

- **Edge TPU Compiler:** Compiles quantized TensorFlow Lite models into a format executable by the Edge TPU (Google Coral, n.d.-b).

Use Cases

- **Real-Time Inference:** High-throughput applications like object detection and image classification (Bi et al., 2019).
- **Distributed AI Processing:** Scalable deployment in edge servers and IoT gateways (Satyanarayanan, 2017d).

Intel Movidius Myriad

The Intel Movidius Myriad is a series of vision processing units (VPUs) designed for high-performance, low-power AI applications (Intel, n.d.-a).

Key Features

- **Neural Compute Engine:** Dedicated hardware blocks for deep learning inference (Moloney, 2014).
- **Power Efficiency:** Designed for minimal power consumption, suitable for battery-powered devices (Venkataramani et al., 2020).
- **Form Factors:** Available as Neural Compute Sticks and integrated into devices (Intel, n.d.-b).

Algorithm Support

- **OpenVINO Toolkit:** Provides tools for optimizing and deploying models on Intel hardware (Intel, n.d.-c).
- **Framework Support:** Compatible with models from TensorFlow, Caffe, and MXNet (Intel, n.d.-d).

Use Cases

- **Edge Vision Systems:** Smart cameras, drones, and augmented reality devices (J. Wu et al., 2020).
- **Industrial Automation:** Quality control and defect detection using AI (Jain et al., 2020).

Neural Processing Units (NPUs)

NPUs are specialized hardware accelerators designed specifically for neural network computations (Y.-H. Chen et al., 2017b).

Key Features

- **Optimized Architecture:** Tailored for matrix multiplication and convolution operations common in AI workloads (Han et al., 2017).
- **Integration:** Often integrated into SoCs (System on Chips) for smartphones and edge devices (Simonyan et al., 2017).
- **Vendor-Specific Implementations:** Examples include Apple's Neural Engine, Huawei's Ascend, and Qualcomm's Hexagon DSP (Apple Inc., n.d.; Huawei, n.d.; Qualcomm, n.d.).

Algorithm Support

- **Framework Integration:** Support through SDKs and APIs provided by hardware vendors (X. Zhang et al., 2019).
- **On-Device AI:** Enables complex AI tasks like facial recognition and natural language processing directly on devices (Ashraf et al., 2019).

Use Cases

- **Mobile AI Applications:** Enhanced camera features, voice assistants, and augmented reality (X. Zhang et al., 2019).

- **IoT Devices:** Smart home devices with advanced AI capabilities (Ashraf et al., 2019).

5.7.4 Hardware-Software Co-Design

Hardware-algorithm co-design involves the simultaneous development of hardware and algorithms to achieve optimal performance and efficiency for AI applications on edge devices.

Importance of Co-Design

- **Performance Optimization:** Tailoring algorithms to specific hardware capabilities can significantly improve performance (Cong et al., 2019).
- **Energy Efficiency:** Co-design enables reduction in power consumption by optimizing computational workloads (Sze, Chen et al., 2017).
- **Resource Utilization:** Efficient use of hardware resources like memory and processing units enhances overall system efficiency (Juan et al., 2018).
- **Customization:** Allows for the creation of specialized solutions for specific applications and constraints (Esmaeilzadeh et al., 2013).

Case Studies

Case Study 1: Eyeriss - MIT's Energy-Efficient Neural Network Accelerator Eyeriss is a custom accelerator designed to run deep convolutional neural networks with high energy efficiency (Y.-H. Chen et al., 2016). Eyeriss utilizes a dataflow architecture that minimizes data movement, which is a major source of energy consumption (Emer, 2016). This achieves significant reductions in energy usage compared to general-purpose processors (Y.-H. Chen et al., 2015).

Case Study 2: NVIDIA's Deep Learning Accelerator (NVDLA) NVDLA is an open-source hardware design for deep learning inference acceleration (NVIDIA,

n.d.-e). It supports customizable configurations to balance power, performance, and area (Tirthapura, 2017). It facilitates integration into SoCs for edge devices, allowing for efficient AI computations (NVIDIA, n.d.-g).

Case Study 3: Google's TPU and Quantization Techniques Google's Tensor Processing Unit (TPU) is designed to accelerate machine learning workloads (Jouppi et al., 2017c). It employs quantization strategies that reduce numerical precision to improve performance and efficiency (Horowitz, 2014). It achieves significant speedups and energy savings in data centers and edge applications (Jouppi et al., 2018).

Case Study 4: ARM's Project Trillium ARM's initiative to provide scalable processors and NPUs for machine learning (ARM Ltd., n.d.-g). Trillium combines hardware IP with software libraries optimized for ARM architectures (ARM Ltd., n.d.-e). Trillium enables partners to develop edge devices with advanced AI capabilities efficiently (D. Howard, 2018).

Implications for Future

The process encourages collaboration between hardware engineers and algorithm developers (Cong & Xiao, 2014). Growing importance of co-design as AI models become more complex and edge devices become more ubiquitous (T. Chen et al., 2020c).

5.8. Application Domains and Use Cases

Edge AI has become increasingly significant across various industries, enabling intelligent applications directly on devices with limited computational resources. This section explores several key domains where edge AI algorithms are making substantial impacts.

5.8.1 Computer Vision on the Edge

Computer vision tasks, traditionally requiring substantial computational power, have been adapted for edge devices through optimized algorithms and models.

Object Detection Models

Object detection involves identifying and localizing objects within an image or video frame. YOLOv5 Nano is a lightweight version of the YOLO (You Only Look Once) family designed for edge deployment (Jocher et al., 2020).

YOLOv5 Nano

- **Architecture:**
  - **Simplified Network:** Reduces the number of layers and parameters compared to larger YOLO models.
  - **Efficiency:** Employs depthwise separable convolutions to decrease computational load (Chollet, 2017).
- **Performance:**
  - **Speed:** Capable of real-time detection on devices like smartphones and embedded systems.
  - **Accuracy:** Maintains reasonable detection accuracy despite the reduced model size.

Applications

- **Surveillance Systems:** Real-time monitoring with limited hardware.
- **Autonomous Drones:** Obstacle detection and navigation without cloud dependency.
- **Retail Analytics:** In-store customer behavior analysis with privacy-preserving on-device processing.

Image Classification

Image classification assigns a label to an entire image, identifying the primary object or scene.

MobileNets MobileNets are efficient convolutional neural networks designed for mobile and embedded vision applications (A. G. Howard et al., 2017a).

- **Key Features:**
  - **Depthwise Separable Convolutions:** Reduces computations and model size.
  - **Parameterization:** Uses width and resolution multipliers to adjust model complexity (Sandler et al., 2018).

Applications

- **Healthcare Diagnostics:** On-device analysis of medical images like X-rays.
- **Agriculture:** Crop disease identification using handheld devices.
- **Wildlife Monitoring:** Species recognition in remote sensors.

Facial Recognition

Facial recognition identifies or verifies a person from a digital image.

MobileFaceNets MobileFaceNets are tailored for efficient face recognition on mobile devices (S. Chen et al., 2018).

- **Technical Aspects:**
  - **Lightweight Structure:** Optimized for low computational cost.
  - **High Accuracy:** Maintains performance suitable for practical applications.

Applications

- **Access Control:** Secure authentication for devices or facilities.
- **Personalization:** Tailoring user experiences in apps based on identity.
- **Law Enforcement:** On-site identification with portable devices.

5.8.2 Audio and Speech Processing

Edge AI enables audio processing tasks to be performed locally, reducing latency and preserving privacy.

Keyword Spotting Algorithms

Keyword spotting detects specific words or phrases in audio streams, commonly used to activate voice assistants.

Wake Word Detection Models

- **Small Footprint Models:** Designed to run continuously with minimal resource usage (Warden, 2018).
- **Deep Learning Approaches:** Use of CNNs and RNNs for higher accuracy.

Applications

- **Voice Assistants:** Activation through wake words like "Hey Siri" or "OK Google."
- **Smart Appliances:** Voice-controlled devices in smart homes.
- **Accessibility Tools:** Hands-free operation for users with mobility impairments.

Speech Recognition Models

Speech recognition converts spoken language into text.

Edge-Optimized Models

- **Model Types:**
  - **Compact RNNs:** Reduced-size recurrent networks for sequence modeling (Graves et al., 2013).
  - **End-to-End Models:** Streamlined architectures combining acoustic and language models.

Applications

- **Transcription Services:** On-device dictation for note-taking apps.
- **Command Recognition:** Voice control for electronics without internet reliance.
- **Language Learning:** Interactive pronunciation feedback.

5.8.3 Natural Language Processing

NLP tasks on the edge enable text processing without the need for cloud services.

Text Classification

Text classification categorizes text into predefined classes.

Efficient NLP Models DistilBERT and TinyBERT: These are compressed versions of BERT suitable for edge deployment (Sanh et al., 2019b). These models are used for faster inference and reduced memory footprint.

Applications

- **Spam Filtering:** Local email or message filtering.
- **Content Moderation:** Real-time detection of inappropriate content in messaging apps.
- **Topic Tagging:** Organizing notes or documents on devices.

Sentiment Analysis Sentiment analysis determines the emotional tone behind textual content.

Lightweight Models

- **Word Embeddings:** Simplified representations of words.
- **Shallow Neural Networks:** Reduced layers for faster processing (Y. Kim, 2014).

Applications

- **Customer Feedback:** Analyzing reviews or feedback on devices.
- **Personal Journals:** Providing insights into mood trends.
- **Chatbots:** Enhancing user interactions by understanding sentiment.

Machine Translation Machine translation automates language translation.

On-Device Translation:

- **Compressed NMT Models:** Smaller models for devices (W. Wu et al., 2020).
- **Bilingual Dictionaries:** Augment models with pre-loaded vocabularies.

Applications

- **Travel Aids:** Offline translation apps for travelers.
- **Education:** Language learning tools without internet dependency.
- **Communication Devices:** Assistive technologies for multilingual interactions.

5.8.4 Anomaly Detection and Predictive Maintenance

Edge AI enables real-time monitoring and maintenance in industrial settings.

Time-Series Analysis Algorithms

Analyzing sequential data to detect patterns and anomalies over time.

LSTM Networks LSTM networks capture temporal dependencies in data (H. Zhao et al., 2017). These networks require optimization for real-time edge processing. Autoencoders are unsupervised models that learn normal patterns and detect deviations.

Applications

- **Equipment Monitoring:** Early detection of machinery faults.
- **Environmental Sensors:** Identifying abnormal readings in climate data.
- **Energy Management:** Detecting irregularities in consumption patterns.

Edge Analytics in Industrial IoT Processing and analyzing data at the source within industrial environments. These integrated systems combine hardware and software for localized analytics (K. Zhang et al., 2020). This leads to real-time decision-making and reduced data transmission.

Applications

- **Predictive Maintenance:** Anticipating equipment failures to schedule timely interventions.
- **Process Optimization:** Adjusting operations based on immediate data insights.
- **Supply Chain Management:** Monitoring logistics for efficiency.

5.8.5 Healthcare and Wearable Devices

Edge AI enhances healthcare delivery through personalized and immediate data processing.

Health Monitoring Algorithms

Signal Processing algorithms filter and interpret biosignals such as ECG or EEG (Biswas et al., 2020). These algorithms classify data for health indicators. The applications of these algorithms include continuous monitoring for conditions like hypertension, real-time feedback on exercise performance, fall detection, and emergency alerts.

Personalized Recommendation Algorithms

- **C**ollaborative Filtering suggests items based on user similarity. This algorithm processes data locally on the edge to maintain privacy (Adomavicius & Tuzhilin, 2011).
- **T**he applications for this include personalized meal plans based on activity and preferences, recommending stress-relief activities, and alerts based on user routines.

5.9. Federated Learning and Collaborative Edge AI

Edge devices, characterized by limited computational resources and privacy concerns, can benefit significantly from collaborative learning approaches. Federated Learning (FL) emerges as a paradigm that allows multiple devices to train a shared model collaboratively while keeping the data localized.

5.9.1 Key Concepts

Federated Learning (FL) is a decentralized machine learning approach where multiple devices, such as smartphones or IoT sensors, train a global model using their local data without transferring it to a central server (B. McMahan et al., 2017).

Key Benefits

- **Privacy Preservation:** By keeping data on local devices, FL minimizes the risk of data breaches and complies with data protection regulations (Shokri & Shmatikov, 2015).
- **Reduced Communication Overhead:** Only model updates are shared, significantly lowering network bandwidth usage (Konečný et al., 2016d).
- **Personalization:** Models can be fine-tuned to reflect local data distributions, enhancing performance for specific user groups or regions (Smith et al., 2017).
- **Scalability:** FL can handle a large number of devices, making it suitable for widespread applications like mobile networks (Bonawitz et al., 2017a).

Algorithms for Federated Learning

Implementing FL requires specialized algorithms that manage distributed training, aggregation of model updates, and ensure convergence while considering device heterogeneity and communication constraints.

Federated Averaging: Federated Averaging (FedAvg) is one of the foundational algorithms in FL that combines local stochastic gradient descent (SGD) on each client with a global model averaging step (H. B. McMahan et al., 2016).

Algorithm Steps:

1. **Initialization:** A global model is initialized on the server.
2. **Client Selection:** A subset of devices (clients) is selected in each training round.
3. **Local Training:**
   - Each selected client downloads the current global model.
   - Clients perform local training on their data for a few epochs.
   - Local model updates are computed.
4. **Communication:** Clients send their local model updates (weights or gradients) back to the server.
5. **Aggregation:**
   - The server aggregates the local updates using weighted averaging:

$$w_{\text{global}}^{t+1} = \sum_{k=1}^{K} \left(\frac{n_k}{n} \right) w_k^{t+1} \tag{5.1}$$

     where w_{global}^{t+1} is the updated global model, w_k^t is the model from client k, n_k is the number of samples on client k, and $n = \sum_{k=1}^{K} n_k$.
6. **Iteration:** Steps 2–5 are repeated until convergence.

The advantages of FL include:

- **Efficiency:** Reduces the number of communication rounds by performing multiple local updates before aggregation (Konečný et al., 2016c).
- **Flexibility:** Accommodates different types of models and optimization algorithms.

Secure Aggregation Protocols: Secure Aggregation ensures that individual client updates remain confidential during the aggregation process (Bonawitz et al., 2017c). The following techniques can be employed for secured aggregation:

- **Additive Secret Sharing:**
  - Clients split their updates into random shares and distribute them to other clients.
  - The server aggregates the shares, and the sum reveals the aggregated update without exposing individual contributions (Shamir, 1979).
- **Homomorphic Encryption:**
  - Clients encrypt their updates; the server performs aggregation on encrypted data, decrypting the result only after aggregation (Paillier, 1999).
- **Noise Addition:**
  - Clients add random noise to their updates. The collective noise cancels out during aggregation, preserving the integrity of the aggregated model (Geyer et al., 2017).

Privacy-Preserving Techniques: Beyond secure aggregation, additional privacy-preserving methods are essential to protect sensitive information during federated learning.

- **Differential Privacy:** Differential Privacy (DP) provides a formal framework for quantifying and limiting the privacy risks associated with data analysis (Dwork, 2006).

- **Application in Federated Learning:**
  - *Local Differential Privacy:* Clients add calibrated noise to their updates before sending them to the server, ensuring that the inclusion or exclusion of a single data point has a minimal impact on the output (Abadi et al., 2016a).
  - *Privacy Budget (ϵ):* A parameter that quantifies the privacy loss; smaller values indicate stronger privacy (D. Song et al., 2019).

Advantages:

- **Quantifiable Privacy Guarantees:** Provides mathematical assurances about data protection.
- **Scalability:** Suitable for large-scale federated systems where client data is highly sensitive.

Homomorphic Encryption: Homomorphic Encryption (HE) allows computations to be performed on encrypted data without decryption, ensuring data remains confidential (Gentry, 2009).

- **Types:**
  - *Partially Homomorphic Encryption:* Supports specific operations (addition or multiplication) on encrypted data (ElGamal, 1985).
  - *Fully Homomorphic Encryption:* Enables arbitrary computations but is computationally intensive and less practical for resource-constrained devices (Brakerski & Vaikuntanathan, 2011).
- **Use in Federated Learning:**
  - *Encrypted Model Updates:* Clients encrypt their updates; the server aggregates these without accessing the raw data (M. Kim et al., 2018).
- **Challenges:**

* *Computational Overhead:* HE schemes can be resource-intensive, posing challenges for edge devices with limited processing capabilities (Halevi & Shoup, 2014).
* *Latency:* Increased computation time may affect the timeliness of model updates.

5.9.2 Real-World Applications of Federated Learning

Google Keyboard

Improves predictive typing and autocorrect features without collecting raw typing data from users (B. McMahan & Ramage, 2017). This keyboard utilizes Federated Averaging to train language models on-device, sending only model updates to the server. The outcome is enhanced user experience with privacy preservation.

Apple's Siri and Dictation

Federated Learning enhances voice recognition and natural language understanding while maintaining user privacy (Apple Machine Learning Research, n.d.). On-device processing and federated learning techniques improve models locally.

Healthcare

Collaborative training of diagnostic models across hospitals without sharing patient data (S. Rieke et al., 2020). Hospitals perform local model training on medical images and share encrypted updates, leading to improved diagnostic accuracy and generalization across diverse datasets.

IoT

Predictive maintenance models trained across multiple factories' equipment data (Y. Lu et al., 2021a). Edge devices on machinery collect data and update local

models, contributing to a global model via federated learning. Early detection of equipment failures, reduced downtime, and protection of proprietary data.

5.10. Secure and Privacy in Edge AI Algorithms

As Edge AI becomes increasingly integrated into critical applications, ensuring the security and privacy of these algorithms is paramount. Edge devices are often deployed in unsecured environments and are susceptible to various threats that can compromise the integrity, confidentiality, and availability of AI models and data (Papernot et al., 2016). This section explores the potential threats, types of attacks, and the defense mechanisms essential for securing Edge AI algorithms.

5.10.1 Threat Models for Edge AI

Edge AI systems face unique challenges due to their distributed nature, resource constraints, and exposure to physical tampering (Vorobeychik & Kantarcioglu, 2018).

Key Threats:

- **Physical Access Attacks:** Adversaries may gain direct access to devices, allowing them to extract sensitive data or inject malicious code (Asghar et al., 2020).
- **Model Extraction:** Attackers attempt to replicate or steal the AI model by observing inputs and outputs (Tramer et al., 2016).
- **Data Privacy Breaches:** Sensitive data processed on edge devices can be intercepted or leaked (Shokri et al., 2017).
- **Adversarial Manipulation:** Inputs to AI models can be manipulated to produce incorrect outputs, leading to system failures (Goodfellow et al., 2015b).

5.10.2 Adversarial Attacks on Edge Algorithms

Adversarial attacks exploit vulnerabilities in AI models to cause unintended behavior. These attacks can be particularly harmful in edge environments where immediate responses are critical.

Evasion Attacks

Evasion Attacks involve crafting malicious inputs, known as adversarial examples, that deceive the AI model into making incorrect predictions while appearing benign to humans (Akhtar & Mian, 2018).

Techniques:

- **Fast Gradient Sign Method (FGSM):** Adds perturbations in the direction of the gradient to maximize the loss (Goodfellow et al., 2014).
- **Projected Gradient Descent (PGD):** Iteratively applies small perturbations within a defined norm bound (Madry et al., 2018).

Impact on Edge AI:

- **Autonomous Vehicles:** Misclassification of traffic signs leading to accidents (Eykholt et al., 2018).
- **Security Systems:** Bypassing facial recognition or intrusion detection mechanisms (Sharif et al., 2016).

Poisoning Attacks

Poisoning Attacks involve contaminating the training data to introduce vulnerabilities into the model (Biggio et al., 2012).

Types:

- **Data Poisoning:** Injecting malicious samples into the training dataset to alter the model's behavior (X. Liu et al., 2018).
- **Backdoor Attacks:** Embedding hidden triggers that, when activated, cause the model to output attacker-specified results (Gu et al., 2017).

Impact on Edge AI:

- **Industrial Control Systems:** Compromised models may misinterpret sensor data, leading to malfunction (L. Yang et al., 2019).
- **Healthcare Devices:** Altered diagnostic models could result in incorrect patient assessments (Fredrikson et al., 2015).

5.10.3 Defense Mechanisms

To mitigate these threats, robust defense strategies are essential in the development and deployment of Edge AI algorithms.

Robust Model Training

Implementing training procedures that enhance the model's resilience to adversarial attacks (Carlini & Wagner, 2017).

Techniques:

- **Adversarial Training:** Incorporating adversarial examples into the training process to improve robustness (Kurakin et al., 2017).
- **Regularization Methods:** Applying techniques like dropout and weight decay to prevent overfitting to adversarial patterns (Zagoruyko & Komodakis, 2016).

Benefits:

- **Improved Generalization:** Models become more resilient to unseen perturbations (Szegedy et al., 2013).
- **Enhanced Security:** Reduces the effectiveness of both evasion and poisoning attacks.

Runtime Monitoring

Implementing systems that monitor the AI model's inputs and outputs during operation to detect anomalies (F. Zhang et al., 2019).

Techniques:

- **Anomaly Detection:** Identifying inputs that deviate significantly from the training data distribution (Hendrycks & Gimpel, 2017).
- **Input Sanitization:** Preprocessing inputs to remove potential adversarial perturbations (J. Song et al., 2018).

Benefits:

- **Real-Time Protection:** Immediate detection and response to adversarial inputs (Wong & Kolter, 2018).
- **System Reliability:** Maintains consistent performance even under attack.

5.10.4 Secure Model Deployment

Ensuring that AI models are securely deployed on edge devices is crucial to prevent unauthorized access and tampering.

Secure Boot and Trusted Execution Environments

Secure Boot: A security standard that ensures a device boots using only software that is trusted by the manufacturer (Rührmair et al., 2010).

Trusted Execution Environments (TEEs): Isolated environments within a device that protect sensitive computations and data (Brasser et al., 2018).

Implementations

- **ARM TrustZone:** Provides hardware isolation for secure execution of code (ARM Ltd., 2009).
- **Intel SGX:** Offers enclaves for secure computation on Intel processors (Costan & Devadas, 2016).

Benefits:

- **Integrity Assurance:** Prevents execution of unauthorized code during startup (Winter, 2008).
- **Data Protection:** Safeguards model parameters and sensitive data during processing.

Model Encryption and Obfuscation

Techniques to protect the AI model from reverse engineering and unauthorized access (M. Ren et al., 2021).

Model Encryption: Encrypting model files so they cannot be read or modified without the proper decryption key (Louis et al., 2019).

Obfuscation Techniques: Transforming the model code into a form that is difficult to understand or reverse-engineer (Oblinsky et al., 2020).

Benefits:

- **Intellectual Property Protection:** Safeguards proprietary models from theft (J. Wang & Wang, 2018).

– **Security Enhancement:** Reduces the risk of model tampering and extraction attacks.

5.11. Future Trends and Research Directions

Edge AI continues to evolve, with emerging algorithms and hardware pushing the boundaries of what is possible on resource-constrained devices. This section explores next-generation algorithms, advances in hardware, integration with emerging technologies, and open research challenges that will shape the future of Edge AI.

5.11.1 Next-Generation Edge AI Algorithms

As the demand for more efficient and powerful AI models grows, new types of algorithms are being developed specifically for edge deployment.

Spiking Neural Networks

Spiking Neural Networks (SNNs) are inspired by the biological neurons in the human brain and process information using discrete spikes rather than continuous values (Maass, 1997). Unlike traditional artificial neural networks, SNNs operate on the timing of spikes, making them inherently event-driven and energy-efficient.

Key Features of SNNs

– **Temporal Coding:** Information is encoded in the timing between spikes, enabling precise temporal patterns (Laughlin & Sejnowski, 2003).

– **Asynchronous Processing:** Neurons fire only when a threshold is reached, reducing unnecessary computations (Ponulak & Kasinski, 2011).

– **Bio-Inspired Learning:** Utilizes learning rules like Spike-Timing-Dependent Plasticity (STDP) for synaptic updates (Markram et al., 1997).

Advantages of SNNs

- **Energy Efficiency:** Lower power consumption due to sparse and event-driven processing (Rueckauer et al., 2017).
- **Real-Time Processing:** Suitable for applications requiring immediate responses, such as robotics and sensory systems (Stromatias et al., 2015).

Challenges of SNNs

- **Training Complexity:** Traditional backpropagation is not directly applicable; requires specialized training algorithms (Kaiser et al., 2020).
- **Hardware Requirements:** Effective implementation often depends on neuromorphic hardware not yet widely available (Ambrogio et al., 2018).

Applications of SNNs

- Real-Time Sensor Networks: Environmental monitoring with minimal energy usage.
- Robotics: Adaptive motor control and perception systems (Lichtsteiner et al., 2008).
- Brain-Computer Interfaces: Processing neural signals for medical applications.

Hyperdimensional Computing

Hyperdimensional Computing (HDC), also known as Vector Symbolic Architectures, represents data using high-dimensional vectors (usually in thousands of dimensions) (Rahimi et al., 2016). HDC is inspired by the human brain's ability to process information holistically and is particularly suited for efficient computations on edge devices.

Key Features of HDC

- **High-Dimensional Representations:** Encodes information in large vectors, allowing for robust and distributed representations (Kanerva, 2009).
- **Simple Operations:** Utilizes basic algebraic operations like addition, multiplication, and permutation (Plate, 1995).
- **Error Resilience:** High-dimensional spaces provide tolerance to noise and errors (Sebastian et al., 2019).

Advantages of using HDC

- **Computational Efficiency:** Simple operations reduce computational overhead (N. Wang et al., 2021).
- **Memory Efficiency:** Compact representations enable storage of complex patterns in limited memory (J.-S. Seo et al., 2011).

Challenges of using HDC

- **Algorithm Development:** Requires new approaches for algorithm design and problem-solving (Halfhill, 2013).
- **Integration with Existing Systems:** Bridging the gap between HDC and conventional machine learning models (Shen et al., 2017).

Applications

- Real-Time Classification: Fast and efficient pattern recognition tasks.
- Sensor Data Fusion: Combining data from multiple sensors in IoT devices (Harris et al., 2018).
- Anomaly Detection: Identifying deviations in time-series data with minimal computations.

5.11.2 Advances in Hardware for Edge AI

Emerging hardware technologies are set to revolutionize Edge AI by providing significant improvements in performance and energy efficiency.

Neuromorphic Computing

Neuromorphic computing involves designing hardware that mimics the neuronal structure and functioning of the human brain (George et al., 2015). This approach aims to achieve high computational efficiency and low power consumption by leveraging the event-driven nature of neural processing.

Key Concepts

- IBM's TrueNorth: A neuromorphic chip containing one million neurons and 256 million synapses (Merolla et al., 2014).
- Intel's Loihi: A research chip enabling on-chip learning with spiking neural networks (Davies et al., 2018b).
- BrainScaleS and SpiNNaker: European projects focusing on large-scale neuromorphic systems (Furber et al., 2014).

Advantages

- **Energy Efficiency:** Orders of magnitude lower power consumption compared to traditional CPUs and GPUs (Mead, 1990).
- **Parallel Processing:** Massive parallelism inherent in neuromorphic architectures enhances performance.

Challenges

- **Edge Devices:** Efficient implementation of SNNs for real-time processing (Akopyan et al., 2015).

- Robotics: Adaptive control systems with low power requirements.
- Cognitive Computing: Emulating human-like perception and decision-making processes.

Photonic Processors

Photonic processors utilize light (photons) instead of electrons to perform computations, offering the potential for ultra-high-speed data processing and low energy consumption (Esser et al., 2016).

Key Features

- **High Bandwidth:** Light waves can carry significantly more data than electrical signals (Caulfield & Dolev, 2010).
- **Parallelism:** Optical systems naturally support parallel data processing, enabling simultaneous computations (Miller, 2017).

Advantages

- **Speed:** Operations occur at the speed of light, significantly increasing processing speeds (Feldmann et al., 2019).
- **Energy Efficiency:** Reduced heat generation and lower power consumption compared to electronic circuits (Tait et al., 2017).

Applications

- High-Speed Data Centers: Accelerate AI computations in server infrastructure.
- Edge AI Acceleration: Enable complex AI tasks on edge devices without significant energy costs (Ríos et al., 2019).
- Telecommunications: Enhance signal processing capabilities in networking equipment.

5.11.3 Integration with Emerging Technologies

5G/6G Networks

The deployment of 5G networks and the research into 6G technologies provide higher bandwidth, lower latency, and improved connectivity, enhancing Edge AI capabilities (Shen et al., 2018).

Impacts on Edge AI

- **Reduced Latency:** Enables real-time data processing and decision-making (Y. Wang et al., 2015).
- **Edge Computing Integration:** Facilitates distributed computing architectures where processing is shared between devices and edge servers (Latva-aho & Leppänen, 2019).
- **Network Slicing:** Allows dedicated network resources for specific applications, improving reliability (Mach & Becvar, 2017c).

Applications to AI

- Autonomous Vehicles: Real-time communication between vehicles and infrastructure (Alliance, 2015).
- Augmented Reality (AR) and Virtual Reality (VR): Enhanced user experiences through low-latency interactions.
- Smart Cities: Efficient management of resources and services through interconnected devices (S. Li et al., 2017).

5.11.4 Open Research Challenges

Despite significant advancements in edge AI, several open problems remain for advancing the sector.

Energy Efficiency in Resource-Constrained Environments

One of the foremost challenges in Edge AI is achieving high energy efficiency without compromising performance. Edge devices often operate on limited power sources, such as batteries or energy harvesting systems, making power consumption a critical concern (L. D. Xu et al., 2014). Developing algorithms and hardware that deliver high computational performance at low energy costs is essential.

Techniques such as model compression, quantization, and pruning have been explored to reduce model size and computational requirements (Gubbi et al., 2013; Han et al., 2016c). However, these methods often lead to trade-offs between accuracy and efficiency. Advancements in low-power hardware, such as specialized AI accelerators and neuromorphic chips, offer promising directions but require further optimization and integration (J. K. Lin et al., 2020; Yan et al., 2018). Exploring new materials and device architectures, such as memristors and spintronics, could also contribute to ultra-low-power AI systems (Ielmini & Wong, 2018; Sengupta et al., 2020).

Security and Privacy in Distributed Edge Environments

Ensuring the security and privacy of data processed on edge devices is a significant challenge. Edge devices are susceptible to physical tampering, malware attacks, and data breaches due to their widespread deployment and often unsecured environments (Vorobeychik & Kantarcioglu, 2018). Protecting sensitive information while enabling real-time processing requires novel cryptographic techniques and privacy-preserving algorithms.

Research efforts focused on federated learning and differential privacy aim to mitigate these risks by enabling collaborative learning without centralizing sensitive data (Papernot et al., 2016; Shokri & Shmatikov, 2015). However, these methods still face challenges related to communication overhead, model conver-

gence, and maintaining data utility. Exploring new paradigms for secure AI model training and inference is essential to build trust in Edge AI applications (Gehr et al., 2018; A. Liu et al., 2017).

Standardization and Interoperability

The fragmentation of standards in Edge AI hinders the development and deployment of interoperable systems across various industries and applications. Diverse hardware architectures, communication protocols, and software frameworks create silos that complicate integration and scaling (Bhardwaj et al., 2020). Establishing standardized frameworks for Edge AI deployment can enhance collaboration and innovation.

Research into common data formats, communication interfaces, and benchmarking methodologies is necessary to facilitate interoperability among edge devices and systems (Brasser et al., 2018). Collaborative efforts between academia, industry, and regulatory bodies can drive the establishment of these standards and promote best practices in Edge AI development.

Explainability and Transparency of Edge AI Models

As AI systems become more pervasive in critical applications, the need for explainable and transparent models grows. Users and regulators demand understanding of how AI models make decisions, especially in areas like healthcare, finance, and autonomous vehicles (Gunning, 2017). However, many high-performing models, such as deep neural networks, are inherently opaque, making it challenging to interpret their inner workings.

Developing methods for model interpretability suitable for resource-constrained edge devices is an open research challenge (Arrieta et al., 2020). Techniques like model distillation, saliency maps, and symbolic reasoning have been proposed, but integrating them into edge deployments without significant overhead remains

difficult (Liao et al., 2020; Ribeiro et al., 2016). Balancing explainability with efficiency is essential for trust and compliance.

Scalability in Massive Edge Networks

Scaling AI applications across vast networks of edge devices presents significant technical hurdles. Edge devices vary widely in computational capabilities, network connectivity, and power availability (Y. Mao et al., 2017). Managing these heterogeneous resources efficiently is a complex problem. Network limitations, such as bandwidth constraints and intermittent connectivity, can impede coordination and synchronization necessary for distributed AI tasks (W. Jiang et al., 2020b).

Developing scalable architectures and protocols that can adapt to dynamic network conditions and device capabilities is essential for the future of Edge AI (T. Chen et al., 2020b). Approaches like hierarchical edge computing, fog computing, and distributed ledger technologies are being explored but require further research to handle the complexity and ensure reliability (Y. Lu et al., 2021b; Yi et al., 2015).

Ethical Considerations and Responsible AI

Ethical issues, including bias, fairness, and user consent, pose significant challenges in deploying Edge AI systems. AI models trained on biased data can perpetuate or amplify societal biases, leading to unfair or discriminatory outcomes (Chouldechova & Roth, 2018). Ensuring that AI systems respect user privacy and obtain proper consent for data usage is critical, particularly as edge devices often collect sensitive personal information (Parzen et al., 2017).

Developing guidelines and frameworks for ethical AI, along with technical solutions for bias mitigation and privacy preservation, is an ongoing area of research (Pasquale, 2015). Incorporating ethical considerations into the design and de-

ployment phases is necessary for responsible AI. Additionally, aligning AI development with legal and societal norms requires multidisciplinary collaboration.

Integration with Emerging Technologies

Integrating Edge AI with technologies like 5G/6G networks, Internet of Things (IoT), and blockchain offers immense potential but introduces new challenges. Coordinating between AI algorithms and communication protocols requires interdisciplinary research (Y. Mao et al., 2017; Yi et al., 2015). Issues such as network slicing, quality of service, and latency must be addressed to ensure seamless operation (**xia2017bbds**).

The convergence of AI with technologies like blockchain for secure, decentralized applications is an emerging area that presents both opportunities and challenges (Chouldechova & Roth, 2018). Developing synergistic solutions that leverage the strengths of these technologies while mitigating their weaknesses is critical for the advancement of Edge AI.

Efficient On-Device Training and Adaptation

Training AI models directly on edge devices is desirable for personalization and privacy but is constrained by limited computational resources. Developing efficient on-device training algorithms that can learn from local data without significant energy consumption or latency is a significant challenge (Pan & Yang, 2010). Techniques like incremental learning, few-shot learning, and transfer learning are being explored, but more research is needed to make them practical for edge deployment (Hao et al., 2018; Y. Wang et al., 2015).

Optimizing these methods for edge hardware and integrating them with privacy-preserving techniques is an open research area. Additionally, enabling edge devices to adapt to changing environments and user behaviors in real-time requires novel algorithms and hardware support (Latva-aho & Leppänen, 2019).

Handling Dynamic and Unreliable Environments

Edge devices often operate in dynamic environments with varying conditions, such as fluctuating network connectivity, changing workloads, and physical disturbances (Z. Li et al., 2018). AI models need to be robust to these changes to maintain performance. Developing adaptive algorithms that can handle uncertainty and adjust to environmental variations is an open problem (S. Deng et al., 2020b).

This includes resilience to hardware failures, changes in data distribution (concept drift), and environmental factors like temperature or interference. Techniques like online learning, adaptive control systems, and robust optimization are being investigated to address these challenges (W. Jiang et al., 2020b).

Economic and Regulatory Challenges

Deploying Edge AI at scale involves economic considerations, such as the cost of hardware, development, and maintenance (**pan2010survey**). Additionally, regulatory challenges related to data protection laws, such as GDPR and CCPA, can impact the design and deployment of Edge AI systems (Kokku et al., 2012; H. Liu et al., 2000). Navigating these legal frameworks while delivering economically viable solutions requires multidisciplinary research involving technology, law, and economics.

Strategies for cost reduction, such as using open-source platforms and collaborative development models, are being explored but must be balanced against potential risks and compliance requirements (J. Zhang et al., 2019). Understanding and addressing the economic barriers and regulatory constraints is essential for the sustainable growth of Edge AI.

Chapter 6

Current state of field

6.1. Distributed Compute

6.1.1 Aethir Edge

Overview of Aethir Edge Aethir Edge is the cutting-edge GPU computing device that unlocks unlimited possibilities for AI, gaming, cloud mobile, and token rewards. Powered by Aethir, it provides powerful computing capabilities at the edge, allowing users to access decentralized compute resources and participate in Aethir's ecosystem, earning them ATH tokens (and more!) as a reward.

Key Features of Aethir Edge:

- **High-Performance Computing:** Powered by the Qualcomm Snapdragon 865 chip, Aethir Edge delivers fast processing power and reduced latency, enabling "real-time processing" for AI, gaming, and video streaming applications.
- **Decentralized GPU Cloud:** The device connects to Aethir's distributed GPU cloud, reducing reliance on centralized cloud services and enhancing privacy by processing data closer to the source.

– **DePIN Rewards:** Earn Aethir's native token (ATH) alongside several partner tokens for contributing computing power and more to the network.

What Aethir Edge Does Well:

– **Powerful Edge Computing:** By utilizing the Qualcomm Snapdragon chip, it delivers high performance through rapid data processing at the edge for a variety of applications.

– **Decentralized Infrastructure:** By leveraging Aethir's distributed GPU cloud, Aethir Edge enhances security and privacy while reducing users' reliance on centralized services.

– **Ultra-Low Energy Efficiency:** The Edge is silent, lightweight, and fits on any desk or shelf. At 18-22 watts, powering an Edge costs less than 0.10USD per day in most countries!

6.1.2 Akash Network

Overview of Akash Network Akash Network is an open-source, decentralized cloud computing platform that operates as a peer-to-peer marketplace for cloud resources. It aims to disrupt the traditional cloud computing industry by providing a more affordable, accessible, and secure alternative to centralized cloud providers like Amazon Web Services (AWS), Google Cloud, and Microsoft Azure.

Key Features of Akash Network:

– **Decentralized Cloud Computing:** Akash Network leverages blockchain technology to reduce dependency on centralized cloud providers, offering enhanced security, transparency, and scalability for users' data and transactions.

- **Permissionless Marketplace:** The platform allows anyone with computational resources to become a cloud provider, fostering competition and driving down prices in an open marketplace.
- **Flexible and Secure Deployment:** Developers can easily deploy applications and workloads on Akash, with the native AKT token ensuring the integrity and authenticity of transactions on the network.
- **Staking and Incentive Mechanism:** AKT token holders can participate in the network by staking their tokens, helping to secure the network and earn rewards.
- **Interoperable Ecosystem:** Akash Network is built on the Cosmos SDK, allowing for easy integration with other blockchain networks and enabling cross-chain collaborations.
- **GPU Marketplace for AI Hosting:** Akash's decentralized GPU marketplace provides a cost-effective and scalable solution for AI researchers and developers to access computational resources.

What Akash Network Does Well:

- **Cost Savings:** Akash's decentralized model and competitive marketplace can "reduce cloud computing costs by up to 85 percent" compared to traditional cloud providers.
- **Fully Open source:** All development progresses are transparent on Akash Github. All events are funded by the community.
- **Accessibility:** The permissionless nature of Akash allows anyone to participate in the cloud computing ecosystem, democratizing access to computational resources.
- **Security and Privacy:** By decentralizing cloud infrastructure and using blockchain technology, Akash enhances the security and privacy of user data and transactions.

- **Scalability:** The network's decentralized architecture enables easy scaling of computational resources to meet the demands of various applications and workloads.
- **Sustainability:** Akash's Proof-of-Stake consensus mechanism is claimed to be more energy-efficient than traditional Proof-of-Work systems, making it a more environmentally friendly cloud computing solution.

6.1.3 Bittensor

Overview of Bittensor Bittensor is a decentralized platform designed to transform digital commodities like compute, data, storage, predictions, and machine intelligence into valuable assets through its unique network of subnets. Powered by the TAO cryptocurrency, Bittensor allows users to participate as miners, validators, or subnet owners, driving innovation in decentralized AI and other markets. The platform aims to foster an open, equitable ecosystem where digital commodities are traded efficiently, and participants are rewarded based on their contributions.

Key Features of Bittensor:

- **Decentralized Digital Commodities:** Enables decentralized production and exchange of digital commodities such as compute power, data, and AI models.
- **TAO Cryptocurrency:** Facilitates transactions and incentives within the ecosystem, rewarding participants for their contributions.
- **Subnet Markets:** Specialized subnets create markets for niche digital products, enhancing scalability and specialization.
- **Open-Source Ecosystem:** Provides tools and documentation for developers to build and participate in decentralized digital markets.

What Bittensor Does Well:

- **Democratization of AI:** Lowers barriers for AI development by decentralizing access to resources, fostering broader participation.
- **Incentive Structure:** Rewards high-quality contributions through a competitive, decentralized marketplace.
- **Scalability and Flexibility:** Subnets enable specialized and scalable markets for different digital commodities, improving network efficiency.
- **Innovation-Friendly:** Offers a flexible, open-source environment that encourages experimentation and growth in decentralized digital economies.

6.1.4 io.net

Overview of io.net Founded in 2022 and based in New York, io.net is a decentralized computing network that provides AI solutions by aggregating global GPU resources. By utilizing the Solana blockchain, io.net offers an efficient platform for the development, execution, and scaling of machine learning (ML) applications.

Key Features of io.net:

- **Decentralized Infrastructure (DePIN):** The io.net aggregates GPUs from underutilized sources such as data centers and crypto miners, offering scalable and customizable access to computational power.
- **Cost Efficiency:** "The platform provides GPU access at up to 90percent lower costs" than traditional cloud providers like AWS and Google cloud, making it an attractive option for developers and startups seeking to minimize expenses.
- **Rapid Deployment:** The io.net allows for fast setup and access to GPU clusters, which is ideal for machine learning engineers who need immediate computational resources.

What io.net Does Well:

- **Affordable AI Compute:** io.net offers significantly cheaper access to GPU resources compared to centralized cloud services, making it cost-effective for developers.
- **Scalability:** The platform allows developers to quickly deploy and scale GPU clusters, optimizing resources for AI model training and other ML tasks.
- **Customizable GPU Access:** By using decentralized resources, io.net provides flexibility in accessing scalable GPU infrastructure at a lower cost.

6.1.5 Kaisar Network

Overview of Kaisar Network Founded in 2024, Kaisar Network is revolutionizing the compute landscape by creating a decentralized platform for distributed GPU resources, optimized for AI model training, inference, and beyond. By leveraging blockchain technology, Kaisar provides a secure and scalable environment for individuals and enterprises to rent and contribute underutilized GPU power. This open ecosystem aims to democratize access to AI computation, making it more affordable and efficient for developers, researchers, and businesses alike.

Key Features of Kaisar Network:

- **Decentralized GPU Protocol:** Kaisar Network's core product is a DePIN Protocol for GPUs. It aggregates idle computing resources from enterprises and consumer devices (like MacBooks, PCs, GPUs) and data centers, transforming them into a decentralized, scalable network for AI and high-performance computing tasks.
- **DePinFi (DePIN + DeFi) Yield Optimization:** Kaisar offers extra revenue streams for compute providers and advanced yield optimization by

bringing DeFi to Depin. Users can earn yields by restaking their machines, with yields coming from both Web2 (like AI model training, rendering) and Web3 (such as staking and DeFi protocols), unlocking liquidity and profitability for contributors.

- **Tokenized Incentives (KAI):** GPU providers and other contributors are rewarded in KAI tokens for their participation, which can be used for governance, staking, and other utilities within the ecosystem. The KAI token plays a key role in incentivizing long-term commitment to the network.

What Kaisar Network Does Well:

- **True Accessibility for AI Compute:** Unlike traditional cloud services that are expensive and limited to large enterprises, Kaisar offers decentralized compute resources at competitive prices, including the ability to leverage consumer-grade devices. This lowers the entry barrier for AI developers, researchers, and smaller startups.
- **Innovative Yield Generation:** Through DePinFi, users can restake their machines to optimize yield generation. By combining real-world GPU usage with financial tools from Web3, Kaisar offers contributors an opportunity to monetize their hardware like never before.
- **Scalability and Flexibility:** Kaisar Network's architecture ensures seamless scalability, whether a user requires a small cluster of GPUs for quick experiments or large-scale resources for complex AI models. The platform is designed to adapt to the needs of the users, allowing for flexible computing at various scales.
- **Community-Centric Ecosystem:** Kaisar actively engages its community through initiatives like the Kaisar Genesis NFT collection, Kaisar Questverse, Kaisar ZeroNode Extension (aiming for +1 million users). This community-first approach ensures that users, node operators, and developers

are deeply integrated into the platform's growth and decision-making process.

6.1.6 NetMind Power

Overview of NetMind Power NetMind Power, founded in 2021 in London, is a decentralized AI computing platform launched by NetMind.ai. It allows users to utilize their unused computing resources for collaborative deep learning and AI model development. The platform aims to democratize access to advanced computational resources, making it especially beneficial for researchers, startups, and small businesses as a cost-effective alternative to traditional cloud services.

Key Features of NetMind Power:

- **Cost-Effective AI Development:** NetMind Power leverages a distributed network of computing resources, significantly reducing the costs of AI model training and inference compared to conventional cloud services.
- **Decentralized Collaboration:** The platform encourages users to contribute their idle computing power in exchange for NetMind Token (NMT)
- **Versatile AI Tools:** NetMind Power supports distributed training, model fine-tuning, and deployment options, making it an all-in-one solution for AI practitioners.

What NetMind Power Does Well:

- **Affordability:** By using a distributed network, NetMind Power offers cost-effective access to AI training and inference, allowing a wider range of users to benefit from powerful computational resources.
- **Collaborative Environment:** The platform promotes a community-driven approach to AI development, enhancing resource utilization through user contributions.

– **Innovative Features:** The platform supports advanced features like distributed model training and easy deployment, catering to the needs of AI practitioners.

6.1.7 Nosana

Overview of Nosana Nosana is a decentralized GPU grid platform powered by Solana and the NOS token. It enables users, miners, and businesses to monetize their idle hardware by becoming Nosana Nodes. The platform provides on-demand, cost-effective access to GPU resources for AI inference and other computational tasks, "offering up to 85 percent lower costs compared to traditional cloud providers".

Key Features of Nosana:

– **Decentralized GPU Grid:** Nosana builds a global computing grid by utilizing idle GPUs, allowing participants to contribute their spare computational power in exchange for NOS tokens.

– **Cost-Effective Access:** By leveraging underutilized hardware, Nosana provides GPU access at a fraction of the cost of traditional cloud services.

– **Eco-Friendly Compute Power:** The platform emphasizes sustainability by reducing reliance on energy-intensive data centers and using existing hardware.

– **AI Inference Workloads:** Nosana specializes in AI inference tasks, such as model training and image generation, making it suitable for various AI applications.

– **Blockchain Integration:**Powered by Solana, Nosana uses blockchain technology to secure its network, facilitate decentralized compute sharing, and manage payments with the NOS token.

What Nosana Does Well:

- **Scalability:** Nosana offers a scalable solution for businesses requiring large amounts of GPU power for AI workloads without investing in expensive infrastructure.
- **Accessibility:** Anyone with idle GPUs, from gaming PCs to professional workstations, can contribute to the network, making high-performance computing more accessible.
- **Low Cost:** The decentralized model allows Nosana to provide GPU resources at lower prices, making it attractive for AI developers and enterprises.
- **Incentivized Participation:** The use of blockchain and the NOS token provides a secure and incentivized framework, ensuring consistent resource availability and reliable performance.

6.1.8 GPU.net

Overview of GPU.net GPU.net is a decentralized platform that provides scalable access to GPU computing resources for tasks such as AI development, scientific research, and rendering. The platform allows users to either rent or contribute their idle GPU resources to the network, using blockchain technology to ensure secure and efficient exchanges. GPU.net rewards contributors with GPoints, its native token.

Key Features of GPU.net:

- **Decentralized GPU Marketplace:** GPU.net connects users needing computational power with those who have idle GPU resources, creating a marketplace for scalable access.
- **Incentive Structure:** Contributors are rewarded with GPoints (1 Gpoint = 1 USD) for providing their GPU resources, ensuring a consistent supply

of compute power.

- **Proof of Compute (PoC):** This algorithm ensures fair resource allocation and high-quality task execution by constantly monitoring the network's computational health.
- **Robust Consensus Mechanism:** The GPU chain combines Proof of Work (PoW) and Proof of Stake (PoS), incorporating them into a distinct Proof of Compute (PoC) algorithm. This mechanism secures the network while efficiently allocating computational resources by monitoring task execution and ensuring fair distribution of GPU power.

What GPU.net Does Well:

- **Democratized Access to GPUs:** GPU.net offers high-performance computing to users who may not have their own GPUs, addressing the global GPU shortage.
- **Incentivized Participation:** The GPoints and GPU token effectively motivates users to contribute their idle GPUs, creating a sustainable ecosystem.
- **Low Cost:** The decentralized model allows Nosana to provide GPU resources at lower prices, making it attractive for AI developers and enterprises.
- **User-Friendly Platform:** GPU.net simplifies GPU resource sharing by eliminating the need for extensive technical expertise, opening access to a wider range of users.
- **Competitive Pricing:** The platform offers affordable rates for GPU rentals, making it attractive for AI developers and enterprises.

6.1.9 Prodia Labs

Overview of Prodia Labs Prodia Labs provides an API platform for generating images using Stable Diffusion models. The platform simplifies the process of image generation by offering various models tailored to different use cases, such as anime, photography, and fantasy. Developers can access Prodia's API without needing to manage their own GPU infrastructure, making image generation fast and scalable.

Key Features of Prodia Labs:

- **Fast Image Generation:** Prodia offers rapid image generation with an average time of 2 seconds per request, supported by a network of over 10,000 GPUs.
- **Diverse Model Selection:** The platform supports a wide range of models, including SD 1.4, Anything V4.5 for anime, Analog V1 for photography, and Elldreth's Vivid for versatile image generation.
- **Customization Options:** Users can fine-tune image outputs by adjusting parameters such as CFG Scale, Steps, and Seed, and can use negative prompts to filter out unwanted elements.
- **Easy Integration:** Prodia offers an API that is simple to integrate into any application, allowing developers to generate images programmatically.

What Prodia Labs Does Well:

- **Ease of Use:** Prodia's API makes image generation accessible to users without requiring complex infrastructure management.
- **Variety of Models:** The platform provides a diverse selection of models, making it adaptable for different creative needs, from photorealism to anime.
- **Fast Turnaround:** With a network of over 10,000 GPUs, Prodia can generate images in as little as 2 seconds, ideal for real-time content creation.

6.1.10 Spheron Network

Overview of Spheron Network Spheron Network is pioneering a groundbreaking approach to deploying AI workloads at the edge, significantly reducing overall costs for decentralized training, fine-tuning, and inferencing. We've developed the world's first compute orchestration and marketplace capable of effectively managing workloads on both retail-grade GPUs and data center-grade servers. Complementing this is our innovative tiering system, allowing users to compare pricing, stability, and performance to select the most suitable solutions for their needs.

Key Features of Spheron Network:

- **Community Cloud (Fizz Node):**
  - Easy Installation: The Fizz Node can be easily installed on any machine, exposing its compute capacity to the marketplace.
  - Affordable Compute Power: Users can purchase these compute cores at a fraction of the cost compared to traditional cloud services.
- **Provider Nodes (Secure Cloud):**
  - Idle Compute Monetization: These nodes can be run in data centers or on larger devices, allowing providers to seamlessly sell their idle compute capacity to end users.
  - Service Level Agreements (SLAs): Providers can attach SLAs to their compute services, catering to businesses that require guaranteed performance and reliability.
- **Tiering System:**
  - Stability and Reliability: We've developed a novel design that brings stability and reliability to GPUs and CPUs entering the network by assigning them tiers based on multiple factors—including performance, uptime, bandwidth, and more.

    * On-Chain SLAs: This system enables us to bring SLAs on-chain for Tier 1 and Tier 2 providers, enhancing trust and transparency.
  - **Matchmakers:**
    * Automated Compute Deployment: Introducing a completely new way to automate infrastructure in the compute deployment space.
    * Workload Orchestration: Matchmakers act as compute workload orchestrators, allocating requested resources by navigating the marketplace efficiently.
  - **Slark Nodes:**
    * Trustless Compute Auditors: These nodes can be run by anyone to validate incoming nodes within the Spheron ecosystem.
    * Tiering System Maintenance: They are responsible for maintaining the tiering system, ensuring network integrity and performance standards.

What Spheron Does Well:

- **Stability, Scalability, and Reliability:** Addressing core problems in decentralized infrastructure, Spheron incorporates stability, scalability, and reliability directly into the network architecture.
- **Protocol-Level Integration:** Automation is a key pillar for scaling infrastructure marketplaces, and Spheron has integrated it at the protocol layer for seamless operations. Seamless Edge AI Workload Deployment: Effortless Deployment: Deploy edge AI workloads seamlessly, benefiting from the network's optimized performance and low latency.
- **Developer-Centric Design:** Spheron's design allows anyone to leverage the network—whether you're bootstrapping an LLM network, deploying a single instance of a model, or launching GPU-as-a-Service or Node-as-a-Service.

6.1.11 Together AI

Overview of Together AI Together AI is a platform that simplifies the process of running, fine-tuning, and deploying open-source AI models. It provides services ranging from serverless models to dedicated GPU clusters, enabling developers and enterprises to leverage high-performance AI models without managing the underlying infrastructure.

Key Features of Together AI:

- **Serverless Model Inference:** Provides access to over 100 models through serverless endpoints, making it easy to integrate AI into applications.
- **Custom Model Training:** Together AI supports custom model training with advanced optimization techniques such as FlashAttention-3 for improved training speed and cost efficiency.
- **Dedicated GPU Clusters:** Offers scalable GPU clusters for larger projects, allowing users to fine-tune models and deploy at scale.
- **Comprehensive Model Range:** Supports a variety of models, including Llama, GPT, and image models, with easy-to-use APIs for integration.

What Together AI Does Well:

- **Scalable Infrastructure:** The platform allows users to scale from small projects to large GPU clusters based on project needs.
- **Cost-Efficient Performance:** Together AI optimizes for both cost and performance, offering up to 11x lower cost compared to competitors like GPT-4.
- **Flexible Model Support:** Provides a wide range of models, enabling customization and fine-tuning for domain-specific tasks.

6.2. Training companies

6.2.1 ChainML

Overview of ChainML ChainML is a decentralized machine learning platform that focuses on providing AI and machine learning models through a blockchain-powered network using NVIDIA H100 and H200 GPU clusters. The platform allows users to access, train, and deploy AI models using decentralized resources. ChainML aims to lower the barriers to entry for machine learning development by making computational resources more affordable and accessible through a distributed network.

Key Features of ChainML:

- **Decentralized AI and ML Models:** ChainML offers decentralized machine learning models, allowing developers to train and deploy models on a distributed network of resources.
- **Blockchain Integration:** The platform uses blockchain technology to secure transactions, ensuring transparency and trust in the allocation of computational resources.
- **Tokenized Rewards:** Contributors who provide computational resources are rewarded with ChainML tokens, incentivizing participation in the network.
- **Scalable Infrastructure:** ChainML allows developers to scale AI models by leveraging a global network of decentralized resources.

What ChainML Does Well:

- **Decentralized AI Development:** ChainML offers a decentralized alternative for AI and machine learning development, reducing costs and increasing accessibility for developers.

- **Incentivized Resource Contribution:** The platform rewards contributors with tokens, encouraging a steady flow of computational resources to support AI development.
- **Scalability:** By using a decentralized network, ChainML enables developers to scale their machine learning models more easily compared to traditional cloud providers.

6.2.2 Gensyn

Overview of Gensyn Gensyn is a decentralized platform that allows AI developers to access large-scale distributed GPU resources for AI model training. The platform uses blockchain technology to create a decentralized marketplace for computational resources, where participants can contribute their GPUs to earn rewards. Gensyn aims to democratize access to AI compute power, making it more affordable and accessible for developers and enterprises.

Key Features of Gensyn:

- **Decentralized GPU Marketplace:** Gensyn provides a marketplace where developers can access distributed GPU resources for AI model training at competitive prices.
- **Token Rewards:** Contributors are rewarded with Gensyn's native tokens for providing their idle GPU resources to the network.
- **Scalability and Flexibility:** The platform allows developers to scale their AI models by leveraging the combined computational power of a decentralized network of GPUs.
- **Blockchain Integration:** Gensyn uses blockchain technology to secure transactions and facilitate transparent, decentralized resource allocation.

What Gensyn Does Well:

- **Cost-Effective Compute Access:** Gensyn offers GPU resources at lower prices compared to traditional cloud providers, making it more affordable for developers with large-scale AI workloads.
- **Scalability:** The platform's decentralized nature allows developers to scale AI models by tapping into a global network of GPUs.
- **Incentives for Contributors:** Gensyn rewards contributors with tokens, encouraging participation and ensuring a steady supply of computational resources.

6.2.3 Prime Intellect

Overview of Prime Intellect Prime Intellect is a decentralized AI platform founded in 2023, focused on "democratizing access to computational resources" for AI model training and development. The platform facilitates collaboration among researchers, enabling the development of open-source AI models while bridging the gap between academic research and industry.

Key Features of Prime Intellect:

- **Decentralized Collaboration:** Researchers can collaborate in a decentralized environment, pooling computational resources to develop AI models and promoting community-driven innovation.
- **Cost Efficiency:** Prime Intellect commoditizes computing resources, offering a more affordable alternative to traditional cloud services, especially for academic researchers and smaller developers.

What Prime Intellect Does Well:

- **Collaborative AI Development:** The platform encourages resource sharing among researchers, fostering innovation and making it easier for smaller entities to participate in AI development.

– **Cost-Effective Access:** Prime Intellect provides affordable access to computing resources, reducing the barriers to AI development for smaller organizations and academic researchers.

6.3. Inference companies

6.3.1 Crynux

Overview of Crynux Crynux is the decentralized orchestration layer on edge AI. In a centralized AI environment, data providers, computing power, AI tasks and applications are handled by the same enterprise, such as Google and OpenAI. But in a decentralized environment, where edge data and edge computing are utilized, these participants are from different entities. There's no trust between them. Crynux builds the orchestration layer to help them coordinate on AI tasks in a permissionless, trustless and serverless manner.

Key Features of Crynux:

– **Decentralized computing on edge:** Crynux launched the first live testnet on edge devices for decentralized computing. Miners can just download the app and run it on their devices.

– **AI service for real user needs:** Crynux provides multi-modality model serving, including: text, image, music and video. Moreover, Crynux offers lang-chain compatible workflow deployment that serves real user needs.

– **Federated finetuning:** Crynux supports decentralized fine-tuning with federated data from the community.

– **Edge AI Engine:** empowered by distributed edge inference and fine-tuning, everyone can run AI models with their own devices and their own data

What Crynux Does Well:

- **Permissionless:** Crynux does not keep you away by whitelisting. Everyone can join the network and serve with their own devices, which unblocks supply from billions of devices.
- **Serverless:** Crynux does not have any centralized server to run their protocol, which makes operating cost to be zero.
- **Trustless:** Crynux enables decentralized entities to coordinate on AI tasks trustless by verifying computing results on chain.
- **Pervasive:** Crynux utilizes edge devices for computing, which makes a pervasive AI experience
- **Privacy Preserved:** Crynux utilizes edge data for computing, while protect users' privacy on device.

6.3.2 Exo Labs

Overview of Exo Labs Exo Labs is a platform that enables distributed AI inference by pooling the computational resources of multiple devices. It dynamically partitions AI models across these devices, allowing users to run large models like Llama on consumer hardware.

Key Features of Exo Labs:

- **Dynamic Model Partitioning:** Splits models across multiple devices based on available resources, enabling distributed AI inference without a centralized master-worker architecture.
- **Wide Model Support:** Supports popular AI models, including Llama, and frameworks such as MLX and tinygrad.
- **Device Collaboration:** Devices like smartphones, laptops, and desktops can pool their resources to run models.

- **Peer-to-Peer Architecture:** All devices in the network operate equally, without reliance on a single controller.

What Exo Labs Does Well:

- **Hardware Agnostic:** Exo supports various device types, from smartphones to desktops, making it highly accessible to users with different hardware configurations.
- **Cost-Effective:** By using existing consumer devices, Exo eliminates the need for expensive dedicated GPUs.
- **Scalability:** The peer-to-peer architecture allows the network to scale naturally as more devices are connected.

6.3.3 HyperspaceAI

Overview of HyperspaceAI HyperspaceAI is a decentralized protocol that enables distributed AI model inference across a global network of nodes, allowing users to explore and interact with over 1,000 AI models. The protocol focuses on democratizing AI access by allowing users to contribute their computational resources for AI tasks like LLM inference. HyperspaceAI operates in a permissionless environment and uses cryptographic techniques for security and fraud prevention.

Key Features of HyperspaceAI:

- **Distributed Model Inference:** HyperspaceAI enables the distribution of AI inference tasks across a decentralized network of nodes.
- **Decentralized Hash Tables (DHTs):** The protocol leverages DHTs for decentralized data storage and efficient node lookup.
- **Incentive Mechanism:** Nodes are rewarded for contributing computational resources, ensuring a consistent supply of compute power.

- **Fraud-Proof Mechanism:** If conflicting results arise, a fraud-proof challenge ensures the correctness of AI outputs, with penalties for incorrect computations.
- **Security and Identity:** Proof of Work (PoW) and public key cryptography are used to secure node identities and prevent attacks.

What HyperspaceAI Does Well:

- **Decentralized Access:** HyperspaceAI democratizes AI access by decentralizing computational power, allowing smaller entities and individuals to participate.
- **Incentivized Participation:** The dynamic reward system encourages nodes to participate actively, ensuring enough resources are available for AI tasks.
- **Strong Security:** The use of advanced cryptographic techniques ensures the security and integrity of the network, making it resistant to common attacks.
- **Scalability:** The distributed nature of the network enables scalability, allowing HyperspaceAI to handle large amounts of computational tasks across multiple nodes.

6.3.4 Infera

Overview of Infera Infera Network is a decentralized AI inference platform that harnesses the latent power of idle GPUs globally, creating a cost-efficient and scalable solution for AI developers. By decentralizing the process of AI inference, particularly for large language models (LLMs), Infera allows anyone with spare computational power to participate as a node runner, earning rewards in the form of INFER tokens. Through its API, developers gain access to a wide

library of open-source AI models, making it easier and cheaper to deploy AI-powered applications.

Key Features of Infera:

- **Decentralized Inference Network:** Utilizes idle consumer GPUs to perform AI inference tasks, reducing the cost of computation significantly.
- **Inference API:** Provides an API for developers to access open-source and custom fine-tuned models on the decentralized network.
- **Token-Based Rewards:** Contributors of idle GPU resources earn tokens based on their participation in AI inference tasks.
- **Flexible Hardware Support:** Supports a variety of GPUs, including Nvidia RTX and AMD cards, making it more accessible to everyday users.
- **Node Participation:** Anyone can become a node runner and earn INFER tokens by contributing GPU power.

What Infera Does Well:

- **Cost Efficiency:** By leveraging consumer-grade GPUs, Infera reduces the costs associated with AI inference compared to centralized providers like Nvidia H100s.
- **API Accessibility:** Developers can easily integrate the platform through APIs compatible with OpenAI's REST standards, simplifying adoption.
- **Community-Driven:** The open-source nature of the platform encourages community participation and innovation.

6.3.5 Kuzco

Overview of Kuzco Kuzco founded in 2024, operates a largely distributed GPU cluster built on the Solana blockchain. The platform facilitates the efficient

and cost-effective inference of large language models (LLMs) by utilizing idle compute resources contributed by network participants, who are rewarded with Kuzco's native token.

Key Features of Kuzco:

- **Distributed GPU Cluster:** Kuzco harnesses a global network of idle GPUs, creating a cohesive and decentralized AI inference infrastructure.
- **Solana Integration:** Built on the Solana blockchain, Kuzco benefits from its high-performance, low-latency, and cost-effective infrastructure.
- **OpenAI-Compatible API:** Developers can easily integrate popular LLMs such as Llama3 and Mistral through the platform's OpenAI-compatible API.
- **Rapid Network Growth:** Kuzco's GPU workers have grown from 467 in March to 8,500 by the end of July 2024, showcasing fast adoption.
- **Hardware Support:** The platform supports multiple hardware configurations across Mac, Windows, and Linux systems, allowing broad participation from different types of devices.

What Kuzco Does Well:

- **Scalability and Accessibility:** Kuzco allows developers to access powerful models like Llama3 and Mistral easily, while expanding its GPU network rapidly.
- **Cost Efficiency:** By using idle GPUs from participants, Kuzco significantly reduces the cost of AI inference tasks compared to centralized alternatives.
- **Global Coverage:** With nodes spread across 70+ countries, Kuzco has established a broad decentralized infrastructure for AI workloads

6.3.6 Lumino

Overview of Lumino Lumino is a decentralized platform designed to provide scalable, AI inference by leveraging a global network of idle GPUs. The platform aims to make AI inference more accessible and affordable by decentralizing computational resources. Lumino focuses on providing infrastructure for inference tasks like natural language processing (NLP), image recognition, and other AI workloads that require significant computational power.

Key Features of Lumino:

- **Decentralized GPU Network:** Lumino connects a global network of GPUs, allowing participants to contribute their idle resources for AI inference tasks.
- **Cost-Effective AI Inference:** By using idle GPUs, Lumino offers AI inference at a lowered cost compared to traditional cloud providers.
- **AI Model Support:** Lumino supports a wide range of AI models, enabling developers to perform tasks such as NLP, image recognition, and more.
- **Tokenized Incentives:** Contributors earn Lumino tokens as rewards for providing their GPUs for AI inference tasks.

What Lumino Does Well:

- **Affordable AI Inference:** By decentralizing GPU resources, Lumino significantly reduces the cost of AI inference, making it more accessible for developers and businesses.
- **Scalable Infrastructure:** The platform allows developers to scale AI models by tapping into a global network of GPUs, ensuring flexibility for various AI tasks.

– **Incentives for Contributors:** The tokenized reward system encourages users to contribute their idle GPUs, ensuring a continuous supply of computational power.

6.3.7 Pin AI

Overview of Pin AI Pin AI is a decentralized platform transforming billions of mobile devices into AI-powered nodes within a global network. It enables the processing of "quadrillions of cross-platform data streams, making personal AI more efficient and accessible". The platform focuses on on-device AI model inference, personal data management, and secure decentralized cloud storage, while ensuring data privacy and user autonomy. By leveraging both edge and cloud-based AI, Pin AI claims to provide scalable and cost-effective solutions for tasks such as data annotation, system optimization, and personalized task execution.

Key Features of Pin AI:

– **On-Device LLM OS:** Provides personal AI services through private, on-device large language models (LLMs), ensuring that data is processed locally to enhance privacy.

– **AI-Powered Data Annotation:** Uses AI models for efficient data labeling, allowing businesses to access high-quality, cost-effective annotation services through a decentralized network.

– **Decentralized Data and Compute Network:** A permissionless network enables distributed AI inference and storage, combining edge AI with cloud computing to maximize value and protect data privacy.

– **Router to Open AI Ecosystem:** Directs complex tasks to external AI services in a permissionless ecosystem, enhancing the platform's flexibility and capability.

- **Data Connector:** Seamlessly retrieves and connects personal data from various apps, creating personalized AI-powered experiences while safeguarding user privacy.
- **Proof of Engagement (PoE) Protocol:** Rewards users for data and activity engagement within the platform's ecosystem, while authenticating interactions securely through cryptographic techniques.

What Pin AI Does Well:

- **Efficient and Private AI Processing:** By leveraging on-device models and decentralized cloud infrastructure. Pin AI ensures data privacy while offering fast and efficient AI-driven task execution.
- **Scalable, Cost-Effective Data Annotation:** Pin AI's decentralized network of AI contributors reduces costs for businesses needing large-scale data labeling.
- **Monetization for Contributors:** Developers can monetize their AI models by contributing to the platform, earning rewards through tasks like data annotation and inference.
- **Open Ecosystem and Flexibility:** Pin AI enables integration with external AI services, expanding its functionality through a permissionless and flexible ecosystem.

6.3.8 Stable Edge

Overview of Stable Edge Stable Edge is a decentralized cloud computing platform designed to support the development of generative AI for small language models, offering high-performance computing resources for AI, machine learning, and big data applications. The platform enables users to contribute their idle computational power and provides a marketplace where developers can access scalable compute resources at lower costs.

Key Features of Stable Edge:

- **Decentralized Cloud Computing:** Stable Edge connects idle compute resources from participants around the world, offering a decentralized alternative to traditional cloud services.
- **Cost-Effective Resource Access:** The platform provides access to high-performance computing power at significantly lower prices compared to conventional cloud providers.
- **AI and Big Data Focus:** Stable Edge specializes in supporting AI, machine learning, and big data workloads, making it a suitable option for developers working on resource-intensive projects.
- **Resource Marketplace:** Users can buy and sell compute power through the Stable Edge marketplace, creating a flexible and dynamic environment for resource allocation.

What Stable Edge Does Well:

- **Lower Costs:** By leveraging decentralized compute power, Stable Edge offers compute resources at much lower prices than traditional cloud providers, making it more accessible for AI developers.
- **Scalability:** The platform provides a scalable solution for users with growing compute needs, particularly for large-scale AI and data processing tasks.
- **Resource Marketplace:** Stable Edge's marketplace model allows for flexible pricing and resource allocation based on supply and demand.

6.3.9 Inference Labs

Overview of Inference Labs Inference Labs provides a platform that delivers AI hyperscale solutions on decentralized networks. By providing secure, scalable, and cryptographically verified AI infrastructure, the platform supports large-scale AI inference tasks. Inference Labs focuses on ensuring certainty in on-chain

AI through decentralized networks, cryptographic proofs, and interoperability across AI systems. Its solutions are tailored to the needs of data scientists, developers, and enterprises looking to deploy next-gen proprietary AI models without compromising on security or scalability.

Key Features of Inference Labs:

- **Decentralized AI with Cryptographic Integrity:** The platform leverages cryptographic verification methods to ensure computational integrity in AI models, allowing for decentralized governance and transparency in AI operations.
- **Mathematically Verifiable Proofs:** Inference Labs ensures AI tasks are secure and verifiable through cutting-edge cryptographic protocols, offering reliability over traditional trust models.
- **AI and Big Data Focus:** Stable Edge specializes in supporting AI, machine learning, and big data workloads, making it a suitable option for developers working on resource-intensive projects.
- **Interoperable AI Intelligence:** The platform is designed to integrate with existing AI protocols and allows cross-chain applications, enabling secure and atomic execution of AI workflows.
- **Ethical and Open-Source Protocols:** Inference Labs promotes a market-driven approach to AI governance through game theory, ensuring that AI advancements remain ethical and transparent.

What Inference Labs Does Well:

- **AI Security and Integrity:** By using cryptographic protocols similar to HTTPS/TLS, the platform ensures AI inference security at scale, making it a pioneer in AI governance and verification.

- **Decentralized AI Ownership:** Empowers participants by providing transparent and secure decentralized AI infrastructure, fostering participation in the AI ecosystem.
- **Scalability for Enterprises:** Inference Labs supports large-scale AI tasks, making it an ideal solution for enterprises and developers needing robust and scalable AI infrastructure.

6.4. Data and Security

6.4.1 DATS Project

Overview of DATS Project The DATS Project is a decentralized cybersecurity platform that leverages DePIN (Decentralized Physical Infrastructure Networks) technology to provide enhanced security solutions. It focuses on building a secure ecosystem where businesses and users can benefit from decentralized infrastructure, reducing dependency on traditional, centralized security mechanisms. DATS aims to revolutionize the cybersecurity industry by enabling secure, cost-effective, and scalable solutions.

Key Features of DATS Project:

- **Decentralized Security Infrastructure:** DATS uses a decentralized network of nodes to offer robust cybersecurity solutions, minimizing single points of failure.
- **Incentivized Security Providers:** Users can contribute their infrastructure or expertise to the network and earn rewards for providing security services.
- **Real-time Threat Detection:** The platform provides decentralized real-time monitoring and threat detection, ensuring rapid responses to emerging cyber threats.

What DATS Project Does Well:

- **Enhanced Security through Decentralization:** By decentralizing the cybersecurity infrastructure, DATS ensures a more resilient and secure network.
- **Cost-Efficiency:** The decentralized approach reduces the costs associated with maintaining centralized security infrastructure.
- **Community-Driven Innovation:** DATS promotes a collaborative environment where security experts and developers can contribute and share solutions, fostering innovation in the cybersecurity space.

6.4.2 Masa

Overview of Masa Masa is a decentralized AI data network revolutionizing how AI developers access high-quality, real-time data. Positioning itself as the decentralized Scale AI, Masa is building essential data infrastructure for future AI development through a global network of miners and validators. Launched with a viral 17-minute CoinList Launchpad sale, Masa is backed by prominent investors like Digital Currency Group, Anagram, and Animoca. The project operates Bittensor Subnet 42 (SN42), forming a crucial part of Masa's scalable infrastructure.

Key Features of Masa:

- **Diverse Data Sources:** Scrapes and structures data from X-Twitter, Discord, Telegram, web pages, and speech-to-text content, providing a rich dataset for AI training.
- **Intelligent Data Scoring and Rewards:** Sophisticated algorithms evaluate workers based on data quality and volume, using statistical analysis and kurtosis-based scoring.

- **Data Quality Assurance:** Employs cosine similarity assessments and continuous optimization to ensure high-quality data.
- **Scalable Infrastructure:** Operates Bittensor Subnet 42, building infrastructure for future AI applications.

What Masa Does Well:

- **Rapid Network Expansion:** Rapidly growing ecosystem of 48,000+ global node workers and 15+ sophisticated institutional validators, creating a robust, decentralized data infrastructure.
- **Proven Data-Product-Market Fit:** Cultivates a thriving community of 100+ AI developers building cutting-edge applications, delivering mission-critical data that fuels innovation in AI development.
- **Pioneering Institutional Adoption:** Spearheads the adoption of decentralized AI infrastructure across both web3 and web2 sectors, with notable traction among leading web2 institutions.
- **Innovative Dual-Token Incentives:** Introduces the first live token in the Bittensor ecosystem, featuring an innovative TAO-MASA dual-token model. This creates a fair, dynamic reward system that incentivizes consistent, high-quality contributions, aligning stakeholder interests across the network.

6.4.3 Ringfence

Overview of Ringfence Ringfence is a decentralized protocol which aggregates and structures datasets, screens against existing data for conflicts, and establishes provenance.The system ensures that no new additions conflict with existing data and allows original data owners to receive compensation when their data is used for AI model training or fine-tuning.

Key Features of Ringfence:

- **IP Conflict Detection System:** Ringfence ensures all data is screened for intellectual property conflicts, establishing ownership and tracking data usage history so originators are fairly compensated.
- **Data Provenance Verification:** Ringfence's offchain system tracks and verifies the provenance of data, ensuring transparency and integrity across multiple ecosystems
- **Compensation Management for Data Usage:** Ringfence's compensation management system calculates and distributes payments to original data owners when their data is used for AI training.
- **API Integration for Web3 Platforms:** Ringfence offers seamless integration with Web3 platforms with a powerful API, enabling offchain data management services for blockchain ecosystems.

What Ringfence Does Well:

- **Fair Data Compensation:** Tracks data usage in real time, automatically compensating creators when their data is used in AI training or fine-tuning.
- **Data Provenance and Conflict Detection:** Via IP conflict screening and data provenance, Ringfence provides robust protection against unauthorized data usage and tracks data used in AI models.
- **Scalable and Integrated Solution:** Ringfence provides a robust API, allowing seamless integration for Web3 platforms for scalable data management, monetization and data provenance.
- **Customizable Compliance Tools:** Ringfence includes tools that help businesses meet regulatory requirements, ensuring compliance for those managing data custody and usage.

Bibliography

Abadi, M., et al. (2016a). Deep learning with differential privacy. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 308–318.

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., & Zheng, X. (2016b). Tensorflow: A system for large-scale machine learning. *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 265–283.

Abolfazli, S., Sanaei, Z., Ahmed, E., Gani, A., & Buyya, R. (2014). Cloud-based augmentation for mobile devices: Motivation, taxonomies, and open challenges. *IEEE Communications Surveys & Tutorials*, 16(1), 337–368.

Abomhara, M., & Køien, G. M. (2015). Cyber security and the internet of things: Vulnerabilities, threats, intruders, and attacks. *Journal of Cyber Security and Mobility*, 4(1), 65–88.

Abouelmehdi, K., Beni-Hessane, A., & Khaloufi, H. (2018). Big healthcare data: Preserving security and privacy. *Journal of Big Data*, 5(1), 1–18.

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6), 734–749.

Adomavicius, G., & Tuzhilin, A. (2011). Context-aware recommender systems. In *Recommender systems handbook* (pp. 191–226).

Agency, I. E. (2021a). Data centres and data transmission networks.

Agency, I. E. (2021b). Data centres and data transmission networks. https://www.iea.org/reports/data-centres-and-data-transmission-networks

Ahmad, M., & Lee, S. P. (2020). Internet of things (iot) enabled smart autonomous vehicles: A review. *IEEE Access*, 8, 117142–117164. https://doi.org/10.1109/ACCESS.2020.3004390

Akhtar, N., & Mian, A. (2018). Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access*, 6, 14410–14430.

Akopyan, F., et al. (2015). Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 34(10), 1537–1557.

Alaa, M., Zaidan, A. A., Zaidan, B. B., Talal, M., & Kiah, M. L. M. (2017). A review of smart home applications based on internet of things. *Journal of Network and Computer Applications*, 97, 48–65. https://doi.org/10.1016/j.jnca.2017.08.017

Alam, M. R., Reaz, M. B. I., & Ali, M. A. M. (2012). A review of smart homes—past, present, and future. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6), 1190–1203.

Aldridge, I. (2013a). *High-frequency trading: A practical guide to algorithmic strategies and trading systems*. John Wiley & Sons.

Aldridge, I. (2013b). *High-frequency trading: A practical guide to algorithmic strategies and trading systems*. John Wiley & Sons.

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials*, 17(4), 2347–2376.

Alibaba Cloud. (2020). Link iot edge.

Alliance, N. G. M. N. (2015). 5g white paper. *Next Generation Mobile Networks Alliance*, 1, 1–125.

Alrawais, A., Alhothaily, A., Hu, C., & Cheng, X. (2017a). Fog computing for the internet of things: Security and privacy issues. *IEEE Internet Computing*, 21(2), 34–42. https://doi.org/10.1109/MIC.2017.37

Alrawais, A., Alhothaily, A., Hu, C., & Cheng, X. (2017b). Fog computing for the internet of things: Security and privacy issues. *IEEE Internet Computing*, 21(2), 34–42. https://doi.org/10.1109/MIC.2017.37

Amazon. (2021). Meet astro, a home robot unlike any other.

Amazon Developer Services. (2021). Alexa voice service integration for aws iot core.

Ambrogio, S., et al. (2018). Equivalent-accuracy accelerated neural-network training using analog memory. *Nature*, 558(7708), 60–67.

Ammar, M., et al. (2018). Internet of things: A survey on the security of iot frameworks. *Journal of Information Security and Applications*, 38, 8–27.

Anthes, C., et al. (2016). State of the art of virtual reality technology. *2016 IEEE Aerospace Conference*, 1–19.

Apache TVM. (2024). An open source machine learning compiler stack for cpus, gpus, and accelerators [Online]. https://tvm.apache.org/

Apple Inc. (n.d.). Apple Neural Engine [[Online; accessed 29-Sep-2024]].

Apple Inc. (2020a). A14 bionic: A new level of performance and power efficiency.

Apple Inc. (2020b). A14 bionic: A new level of performance and power efficiency [Retrieved from https://www.apple.com/newsroom/2020/10/iphone-12-and-iphone-12-mini-a-new-era-for-iphone-with-5g/].

Apple Inc. (2020c). Iphone 12 pro and iphone 12 pro max: The most powerful iphones ever with advanced technologies.

Apple Inc. (2020d). Siri learning guide.

Apple Inc. (2021a). Advanced privacy technologies.

Apple Inc. (2021b). Privacy-preserving machine learning.

Apple Machine Learning Research. (n.d.). Federated Learning [[Online; accessed 29-Sep-2024]].

ARM Limited. (2013). Big.little technology: The future of mobile.

ARM Ltd. (n.d.-a). ARM Cortex-A Series [[Online; accessed 29-Sep-2024]].

ARM Ltd. (n.d.-b). ARM Cortex-M Series [[Online; accessed 29-Sep-2024]].

ARM Ltd. (n.d.-c). CMSIS-NN: Neural Network Kernels for Cortex-M CPUs [[Online; accessed 29-Sep-2024]].

ARM Ltd. (n.d.-d). Compute Library [[Online; accessed 29-Sep-2024]].

ARM Ltd. (n.d.-e). Ethos-N NPU Series [[Online; accessed 29-Sep-2024]].

ARM Ltd. (n.d.-f). NEON Intrinsics [[Online; accessed 29-Sep-2024]].

ARM Ltd. (n.d.-g). Project Trillium: Machine Learning [[Online; accessed 29-Sep-2024]].

ARM Ltd. (2009). Arm security technology building a secure system using trustzone technology.

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., et al. (2010a). A view of cloud computing. *Communications of the ACM*, 53(4), 50–58. https://doi.org/10.1145/1721654.1721672

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., et al. (2010b). A view of cloud computing. *Communications of the ACM*, 53(4), 50–58. https://doi.org/10.1145/1721654.1721672

Arrieta, S. B., et al. (2020). Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58, 82–115.

Arvin, F., Samsudin, K., & Turgut, A. E. (2014a). Development of an autonomous micro robot for swarm robotics. *International Journal of Advanced Robotic Systems*, 11(3), 42.

Arvin, F., Samsudin, K., & Turgut, A. E. (2014b). Development of an autonomous micro robot for swarm robotics. *International Journal of Advanced Robotic Systems*, 11(3), 42.

Asanović, K., & Patterson, D. (2014). *Instruction sets should be free: The case for risc-v* (tech. rep. No. UCB/EECS-2014-146). EECS Department, UC Berkeley.

Asghar, A., et al. (2020). Security and privacy in mobile edge computing: Challenges and solutions. *IEEE Communications Surveys & Tutorials*, 22(1), 212–249.

Ashraf, M. I., et al. (2019). Edge intelligence for internet of things: A feasibility study. *IEEE Internet of Things Journal*, 6(4), 7192–7200.

Azuma, R. (1997). A survey of augmented reality. *Presence: Teleoperators & Virtual Environments*, 6(4), 355–385.

Azure, M. (2021). Azure ai platform. https://azure.microsoft.com/services/machine-learning/

Ba, J., & Caruana, R. (2014). Do deep nets really need to be deep? *Advances in Neural Information Processing Systems (NeurIPS)*.

Bachem, O., Lucic, M., et al. (2017). Practical coreset constructions for machine learning. *Journal of Machine Learning Research*.

Badue, C., et al. (2021a). Self-driving cars: A survey. *Expert Systems with Applications*, 165, 113816.

Badue, C., Guidolini, R., Carneiro, R. V., Azevedo, P., Cardoso, V. B., Forechi, A., & Oliveira-Santos, T. (2021b). Self-driving cars: A survey. *Expert Systems with Applications*, 165, 113816. https://doi.org/10.1016/j.eswa.2020.113816

Baltrusaitis, T., Ahuja, C., & Morency, L. P. (2019). Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2), 423–443.

Banbury, C. R., Reddi, V. J., Lam, M., Fu, W., Fazel, A., Holleman, J., et al. (2020). Benchmarking tinyml systems: Challenges and direction. *Proceedings of the 2020 Conference on Machine Learning and Systems*, 8–17.

Banbury, C. R., Reddi, V. J., Lam, M., et al. (2020). Benchmarking tinyml systems: Challenges and direction. *Proceedings of the 2020 Conference on Machine Learning and Systems*, 8–17.

Banbury, C. R., Reddi, V. J., Lam, M., Fu, W., Fazel, A., Holleman, J., & Whatmough, P. N. (2020). Benchmarking tinyml systems: Challenges and direction. *Proceedings of the 2020 Conference on Machine Learning and Systems*, 8–17.

Batty, M. (2018). Digital twins. *Environment and Planning B: Urban Analytics and City Science*, 45(5), 817–820.

Bengio, Y., et al. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Benke, K., & Tomkins, B. (2017). Future food-production systems: Vertical farming and controlled-environment agriculture. *Sustainability: Science, Practice and Policy*, 13(1), 13–26.

Ben-Sasson, E., Chiesa, A., Genkin, D., Tromer, E., & Virza, M. (2014). Snarks for c: Verifying program executions succinctly and in zero knowledge. *Advances in Cryptology–CRYPTO 2013*, 90–108.

Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9), 509–517.

Bhardwaj, K., et al. (2020). Mems-based smart sensors and edge computing platforms for iot applications. *IEEE Transactions on Industrial Informatics*, 16(4), 2425–2433.

Bhattacharya, D., & Pal, S. (2015). Anomaly detection: A survey. *International Journal of Computer Applications*, 116(9), 1–8.

Bi, S., et al. (2019). Edge intelligence: Paving the last mile of artificial intelligence with edge computing. *Proceedings of the IEEE*, 107(8), 1738–1762.

Bickmore, T., & Picard, R. (2005). Establishing and maintaining long-term human-computer relationships. *ACM Transactions on Computer-Human Interaction*, 12(2), 293–327.

Biggio, B., et al. (2012). Poisoning attacks against support vector machines. *Proceedings of the 29th International Conference on Machine Learning*, 1467–1474.

Billinghurst, M., Clark, A., & Lee, G. (2015). A survey of augmented reality. *Foundations and Trends® in Human–Computer Interaction*, 8(2-3), 73–272.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Biswas, S., et al. (2020). Wearable continuous ecg monitoring and real-time heart rate variability analysis with a miniaturized wireless platform. *IEEE Transactions on Biomedical Engineering*, 67(7), 1738–1749.

Bixby, H., & Renaudin, M. (2019a). Understanding latency requirements for consumer mobile applications. *IEEE Consumer Electronics Magazine*, 8(2), 20–25. https://doi.org/10.1109/MCE.2018.2885019

Bixby, H., & Renaudin, M. (2019b). Understanding latency requirements for consumer mobile applications. *IEEE Consumer Electronics Magazine*, 8(2), 20–25. https://doi.org/10.1109/MCE.2018.2885019

Blalock, D., Ortiz, J. J. G., Frankle, J., & Guttag, J. (2020). What is the state of neural network pruning? *Proceedings of Machine Learning and Systems*, 129–146.

Blog, G. A. (2020). On-device machine learning: Federated learning and federated analytics.

Blum, M., Feldman, P., & Micali, S. (1988). Non-interactive zero-knowledge and its applications. *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, 103–112.

Bonawitz, K., et al. (2017a). Practical secure aggregation for privacy-preserving machine learning. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 1175–1191.

Bonawitz, K., et al. (2017b). Practical secure aggregation for privacy-preserving machine learning. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 1175–1191.

Bonawitz, K., et al. (2017c). Secure aggregation for federated learning. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 1175–1191.

Boneh, D., & Lipton, R. J. (1996). Algorithms for black-box fields and their application to cryptography. *Advances in Cryptology—CRYPTO'96*, 283–297.

Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012). Fog computing and its role in the internet of things. *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, 13–16.

Boston Dynamics. (2021). Technology.

Brakerski, Z., & Vaikuntanathan, V. (2011). Efficient fully homomorphic encryption from (standard) lwe. *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, 97–106.

Brasser, F., et al. (2018). Trusted execution environments: A look under the hood. *ACM Computing Surveys*, 51(2), 1–36.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32.

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. CRC Press.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, 1877–1901.

Bryant, R. E., Katz, R. H., & Lazowska, E. D. (2008). Big-data computing: Creating revolutionary breakthroughs in commerce, science, and society.

Buciluǎ, C., Caruana, R., & Niculescu-Mizil, A. (2006). Model compression. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*.

Bünz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., & Maxwell, G. (2018). Bulletproofs: Short proofs for confidential transactions and more. *2018 IEEE Symposium on Security and Privacy*, 315–334.

Burges, C. J. C. (1996). Simplified support vector decision rules. *Proceedings of the 13th International Conference on Machine Learning*, 71–77.

Cai, H., et al. (2019a). Once-for-all: Train one network and specialize it for efficient deployment. *arXiv preprint arXiv:1908.09791*.

Cai, H., Zhu, L., & Han, S. (2019b). Proxylessnas: Direct neural architecture search on target task and hardware. *International Conference on Learning Representations (ICLR)*.

California Legislative Information. (2018). California consumer privacy act (ccpa) of 2018.

Carlini, N., & Wagner, D. (2017). Adversarial examples are not easily detected: Bypassing ten detection methods. *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, 3–14.

Carr, N. (2011). *The shallows: What the internet is doing to our brains*. W. W. Norton Company.

Cassinelli, A., & Ishikawa, M. (2005). Khronos projector. *ACM SIGGRAPH 2005 Emerging Technologies*, 10.

Caulfield, H. J., & Dolev, S. (2010). Why future supercomputing requires optical interconnects. *Nature Photonics*, 4(5), 261–263.

Celio, C., et al. (2017). Boom v2: An open-source out-of-order risc-v core. *First Workshop on Computer Architecture Research with RISC-V (CARRV)*.

Chan, P. W., & Yeung, R. W. (2012). Privacy protection in data mining on mobile devices through data masking. *IEEE Transactions on Knowledge and Data Engineering*, 24(11), 2077–2090.

Chaudhuri, S., Thompson, H., & Demiris, G. (2014). Fall detection devices and their use with older adults: A systematic review. *Journal of Geriatric Physical Therapy*, 37(4), 178–196.

Chen, J., et al. (2016a). Data-driven approach for freeway origin–destination matrix estimation using fusion data from multiple sensors. *Journal of Intelligent Transportation Systems*, 20(3), 275–285.

Chen, J., Ma, T., & Xiao, C. (2018). Fastgcn: Fast learning with graph convolutional networks via importance sampling. *Proceedings of the 6th International Conference on Learning Representations (ICLR)*.

Chen, J., Monga, R., Bengio, S., & Jozefowicz, R. (2016b). Revisiting distributed synchronous sgd. *arXiv preprint arXiv:1604.00981*.

Chen, J., & Ran, X. (2019). Deep learning with edge computing: A review. *Proceedings of the IEEE*, 107(8), 1655–1674. https://doi.org/10.1109/JPROC.2019.2921977

Chen, M., Hao, Y., & Hwang, K. (2018). Real-time data processing at the edge using defi-based resource allocation. *IEEE Network*, 32(1), 73–79.

Chen, M., Mao, S., & Liu, Y. (2014). Big data: A survey. *Mobile Networks and Applications*, 19(2), 171–209. https://doi.org/10.1007/s11036-013-0489-0

Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. d. O., Kaplan, J., et al. (2021). Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*. https://doi.org/10.48550/arXiv.2107.03374

Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. d. O., Kaplan, J., Zaremba, W., et al. (2021). Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*. https://doi.org/10.48550/arXiv.2107.03374

Chen, S., et al. (2018). Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices. *Chinese Conference on Biometric Recognition*, 428–438.

Chen, T., Jin, X., Shen, S., & Han, S. (2019). Learning efficient object detection models with knowledge distillation. *Advances in Neural Information Processing Systems*, 742–753.

Chen, T., Zhang, S., & Li, J. (2020a). End-to-end learning for self-driving cars: An overview of recent advances. *IEEE Transactions on Intelligent Vehicles*, 5(4), 724–735.

Chen, T., et al. (2020b). A survey on lightweight deep learning models for resource-constrained applications. *IEEE Internet of Things Journal*, 7(8), 6174–6195.

Chen, T., et al. (2020c). Hardware accelerators for machine learning. *Synthesis Lectures on Computer Architecture*, 15(2), 1–158.

Chen, T., Zhang, S., & Li, J. (2020d). End-to-end learning for self-driving cars: An overview of recent advances. *IEEE Transactions on Intelligent Vehicles*, 5(4), 724–735.

Chen, W., Wilson, J., Tyree, S., Weinberger, K., & Chen, Y. (2015). Compressing neural networks with the hashing trick. *International Conference on Machine Learning (ICML)*.

Chen, Y., Bellavitis, C., & Chhabra, K. (2020). Blockchain disruption and decentralized finance: The rise of decentralized business models. *Journal of Business Venturing Insights*, 13, e00151.

Chen, Y., & Lin, Z. (2021). Edge ai: Empowering ai at the edge. *IEEE Internet of Things Magazine*, 4(2), 8–9.

Chen, Y., Yu, T., & Xu, Z. (2019). A survey on edge computing systems and tools. *Proceedings of the IEEE*, 107(8), 1537–1562.

Chen, Y.-H., Yang, T.-J., Emer, J. S., & Sze, V. (2017a). Eyeriss v2: A flexible accelerator for emerging deep neural networks on mobile devices. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 9(2), 292–308. https://doi.org/10.1109/JETCAS.2019.2910232

Chen, Y.-H., et al. (2015). A survey of accelerator architectures for deep neural networks. *IEEE Micro*, 35(3), 24–35.

Chen, Y.-H., et al. (2016). Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks. *ACM SIGARCH Computer Architecture News*, 44(3), 367–379.

Chen, Y.-H., et al. (2017b). Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks. *IEEE Journal of Solid-State Circuits*, 52(1), 127–138.

Chen, Y.-H., Yang, T.-J., Emer, J. S., & Sze, V. (2017c). Eyeriss v2: A flexible accelerator for emerging deep neural networks on mobile devices. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 9(2), 292–308.

Chen, Z., Xu, X., & Liu, Z. (2022a). On-device natural language processing: An edge ai perspective. *IEEE Transactions on Neural Networks and Learning Systems*, 33(11), 6136–6153.

Chen, Z., Xu, X., & Liu, Z. (2022b). On-device natural language processing: An edge ai perspective. *IEEE Transactions on Neural Networks and Learning Systems*, 33(11), 6136–6153.

Cheng, B., Yang, J., Xu, Y., & Zhao, W. (2018a). Energy-efficient smart home automation system using edge computing. *IEEE International Conference on Edge Computing (EDGE)*, 62–69. https://doi.org/10.1109/EDGE.2018.00016

Cheng, B., Yang, J., Xu, Y., & Zhao, W. (2018b). Energy-efficient smart home automation system using edge computing, 62–69.

Cheng, B., Yang, J., Xu, Y., & Zhao, W. (2018c). Energy-efficient smart home automation system using edge computing. *IEEE International Conference*

on Edge Computing (EDGE), 62–69. https://doi.org/10.1109/EDGE.2018.00016

Ching, T., et al. (2018). Opportunities and obstacles for deep learning in biology and medicine. *Journal of the Royal Society Interface*, 15(141), 20170387.

Cho, K., et al. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1724–1734.

Choi, J., El-Khamy, M., & Lee, J. (2018). Towards the limit of network quantization. *IEEE Journal of Selected Topics in Signal Processing*, 12(4), 733–748.

Choi, Y., El-Khamy, M., & Lee, J. (2018). Towards the limit of network quantization. *IEEE Journal of Selected Topics in Signal Processing*, 12(4), 733–748.

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1251–1258.

Chouldechova, A., & Roth, A. (2018). The frontiers of fairness in machine learning. *arXiv preprint arXiv:1810.08810*.

Cisco. (2020a). Cisco annual internet report (2018–2023) [Retrieved from https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html].

Cisco. (2020b). Cisco annual internet report (2018–2023).

Cisco. (2020c). Cisco annual internet report (2018–2023).

Cisco Systems. (2018). Cisco visual networking index: Forecast and trends, 2017–2022.

Cloud, G. (2018). Edge tpu overview.

Cloud, G. (2021). Cloud ai products. https://cloud.google.com/products/ai/

Cong, J., et al. (2019). Hardware-software co-design of neural networks for efficient ai computing. *Proceedings of the IEEE*, 107(8), 1413–1432.

Cong, J., & Xiao, B. (2014). Minimizing computation in convolutional neural networks. *International Conference on Artificial Neural Networks*, 281–290.

Coral. (2021). Products.

Coral edge tpu: Supercharging inference at the edge [Accessed: 2024-09-27]. (2024).

Corporation, N. (2021). Nvidia jetson platform.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297.

Costan, V., & Devadas, S. (2016). Intel sgx explained.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1), 21–27.

Creswell, A., et al. (2018). Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*.

Dai, H.-N., Zheng, Z., & Zhang, Y. (2019). Blockchain for internet of things: A survey. *IEEE Internet of Things Journal*, 6(5), 8076–8094. https://doi.org/10.1109/JIOT.2019.2920987

Danks, D., & London, A. J. (2017). Regulating autonomous systems: Beyond standards. *IEEE Intelligent Systems*, 32(1), 88–91.

Davenport, T. H., & Kirby, J. (2016). Just how smart are smart machines? *MIT Sloan Management Review*, 57(3), 21–25.

Davenport, T. H., & Ronanki, R. (2018). Artificial intelligence for the real world. *Harvard Business Review*, 96(1), 108–116.

David, R., Duke, B., Rueckert, D., et al. (2021). Tensorflow lite micro: Embedded machine learning for tinyml systems. *Proceedings of the NeurIPS Workshops*.

Davies, M., et al. (2018a). Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro*, 38(1), 82–99.

Davies, M., et al. (2018b). Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro*, 38(1), 82–99.

Davis, N., et al. (2015). Creativity support tools: Report from a u.s. national science foundation sponsored workshop. *International Journal of Human-Computer Interaction*, 20(2), 61–77.

Dean, J., & Ghemawat, S. (2008). Mapreduce: Simplified data processing on large clusters. *Communications of the ACM*, 51(1), 107–113.

Deng, L., Li, G., Han, S., Shi, L., & Xie, Y. (2020a). Model compression and hardware acceleration for neural networks: A comprehensive survey. *Proceedings of the IEEE*, 108(4), 485–532. https://doi.org/10.1109/JPROC.2020.2976475

Deng, L., Li, G., Han, S., Shi, L., & Xie, Y. (2020b). Model compression and hardware acceleration for neural networks: A comprehensive survey. *Proceedings of the IEEE*, 108(4), 485–532. https://doi.org/10.1109/JPROC.2020.2976475

Deng, S., Zhao, H., Fang, W., Yin, J., Dustdar, S., & Zomaya, A. Y. (2020a). Edge intelligence: The confluence of edge computing and artificial intelligence. *IEEE Internet of Things Journal*, 7(8), 7457–7469. https://doi.org/10.1109/JIOT.2020.2984887

Deng, S., et al. (2020b). Edge intelligence: The confluence of edge computing and artificial intelligence. *IEEE Internet of Things Journal*, 7(8), 7457–7469.

Deng, S., Zhao, H., Fang, W., Yin, J., Dustdar, S., & Zomaya, A. Y. (2020c). Edge intelligence: The confluence of edge computing and artificial intelligence. *IEEE Internet of Things Journal*, 7(8), 7457–7469.

Denton, E., Zaremba, W., Bruna, J., LeCun, Y., & Fergus, R. (2014). Exploiting linear structure within convolutional networks for efficient evaluation. *Advances in Neural Information Processing Systems (NeurIPS)*.

Dinechin, B. D. D., et al. (2013). A clustered manycore processor architecture for embedded and accelerated applications. *High Performance Extreme Computing Conference (HPEC)*, 1–6.

Ding, A. Y., Kousiouris, G., & Mavromoustakis, C. X. (2019a). Edge and fog computing for the internet of things: A survey on current trends and future directions. *IEEE Access*, 7, 111022–111035. https://doi.org/10.1109/ACCESS.2019.2931517

Ding, A. Y., Kousiouris, G., & Mavromoustakis, C. X. (2019b). Edge and fog computing for the internet of things: A survey on current trends and future directions. *IEEE Access*, 7, 111022–111035.

Ding, A. Y., Kousiouris, G., & Mavromoustakis, C. X. (2019c). Edge and fog computing for the internet of things: A survey on current trends and future directions. *IEEE Access*, 7, 111022–111035.

Dong, J., Ni, R., Wang, L., Zha, H., & Huang, W. (2019). Network pruning via transformable architecture search. *NeurIPS*.

Dwork, C., & Roth, A. (2014). The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4), 211–407.

Dwork, C. (2006). Differential privacy. *Automata, Languages and Programming*, 1–12.

Edge Impulse. (n.d.). Edge impulse documentation [Accessed: 2023-10-03].

ElGamal, T. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4), 469–472.

Elsken, T., Metzen, J. H., & Hutter, F. (2019). Neural architecture search: A survey. *Journal of Machine Learning Research*, 20(55), 1–21.

Emer, J. S. (2016). Eyeriss: A tiled architecture for deep convolutional neural networks.

Ericsson. (2020). Ericsson mobility report.

Esmaeilzadeh, H., et al. (2013). Neural acceleration for general-purpose approximate programs. *IEEE Micro*, 33(3), 16–27.

Esser, S. K., et al. (2016). Convolutional networks for fast, energy-efficient neuromorphic computing. *Proceedings of the National Academy of Sciences*, 113(41), 11441–11446.

Esteva, A., et al. (2019). A guide to deep learning in healthcare. *Nature Medicine*, 25(1), 24–29.

European Data Protection Board. (2019). Guidelines 1/2020 on processing personal data in the context of connected vehicles and mobility related applications.

European Parliament and Council of European Union. (2016a). Regulation (eu) 2016/679 (general data protection regulation).

European Parliament and Council of European Union. (2016b). Regulation (eu) 2016/679 (general data protection regulation). *Official Journal of the European Union*, L119, 1–88.

European Union. (2016). General data protection regulation (gdpr). *Official Journal of the European Union*. https://eur-lex.europa.eu/eli/reg/2016/679/oj.

Eykholt, K., et al. (2018). Robust physical-world attacks on deep learning models. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1625–1634.

Fan, A., Bhosale, S., Schwenk, H., Ma, M., El-Kishky, A., Goyal, S., et al. (2021). Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(107), 1–48. https://jmlr.org/papers/v22/20-1307.html

Fan, K., Ren, Y., Wang, Y., & Yang, Y. (2019). Decentralized data storage for edge ai using defi models. *IEEE Transactions on Industrial Informatics*, 15(12), 6513–6522.

Federal Trade Commission. (1998). Children's online privacy protection act (coppa).

Federated learning on blockchain [Retrieved October 2023]. (n.d.). https://www.flock.io/

Feige, U., Fiat, A., & Shamir, A. (1988). Zero-knowledge proofs of identity. *Journal of Cryptology*, 1(2), 77–94.

Feldmann, S., et al. (2019). All-optical spiking neuromorphic networks with self-learning capabilities. *Nature*, 569(7755), 208–214.

Feng, J., & Zhang, W. (2018). Joint resource allocation and incentive design for edge computing. *IEEE Transactions on Wireless Communications*, 17(8), 5445–5457. https://doi.org/10.1109/TWC.2018.2837484

Fitbit News. (2022). Fitbit and google research collaborate to advance personalized health and wellness.

Fredrikson, M., et al. (2015). Model inversion attacks that exploit confidence information and basic countermeasures. *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 1322–1333.

Frey, C. B., & Osborne, M. A. (2017). The future of employment: How susceptible are jobs to computerisation? *Technological Forecasting and Social Change*, 114, 254–280.

Furber, S., et al. (2014). The spinnaker project. *Proceedings of the IEEE*, 102(5), 652–665.

Gai, K., Qiu, M., & Zhao, H. (2017). Privacy-preserving data encryption strategy for big data in mobile cloud computing. *IEEE Transactions on Big Data*, 3(2), 107–119. https://doi.org/10.1109/TBDATA.2016.2638432

Gale, T., Elsen, E., & Hooker, S. (2019). The state of sparsity in deep neural networks. *arXiv preprint arXiv:1902.09574*. https://doi.org/10.48550/arXiv.1902.09574

Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4), 1–37.

Gao, F., & Zhou, Y. (2020). Efficient resource utilization in edge computing through defi. *IEEE Access*, 8, 120915–120927.

Gao, Y., Li, H., Zhang, W., Yang, B., & Shen, J. (2019). A secure and privacy-preserving data aggregation scheme for smart grid. *IEEE Transactions on Industrial Informatics*, 15(9), 4943–4952.

Ge, F., Wang, X., Hu, B., & Chen, X. (2021). Personalized route planning for autonomous vehicles using natural language interface. *IEEE Transactions on Intelligent Transportation Systems*, 22(5), 3048–3058.

Gehr, T., et al. (2018). Deepsec: A uniform platform for security analysis of deep learning model. *IEEE Symposium on Security and Privacy*.

Gentry, C. (2009). Fully homomorphic encryption using ideal lattices. *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, 169–178.

George, J. K., et al. (2015). Neuromorphic photonics with electro-optic nonlinearities. *Optica*, 2(10), 865–871.

Geyer, R. C., et al. (2017). Differentially private federated learning: A client level perspective.

Girshick, R., et al. (2016). Region-based convolutional networks for accurate object detection and segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(1), 142–158.

Goldreich, O. (2009). *Foundations of cryptography: Volume 2, basic applications*. Cambridge University Press.

Goodfellow, I. J., et al. (2015a). Explaining and harnessing adversarial examples. *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.

Goodfellow, I. J., et al. (2014). Explaining and harnessing adversarial examples.

Goodfellow, I. J., et al. (2015b). Explaining and harnessing adversarial examples. *International Conference on Learning Representations*.

Google. (2021). Offline language translation in google translate.

Google. (2023). Edge tpu.

Google Cloud. (2018). Edge tpu overview.

Google Cloud. (2019). Edge tpu performance benchmarks.

Google Coral. (n.d.-a). Coral Dev Board [[Online; accessed 29-Sep-2024]].

Google Coral. (n.d.-b). Edge TPU Compiler [[Online; accessed 29-Sep-2024]].

Google Coral. (n.d.-c). Edge TPU Model Compatibility [[Online; accessed 29-Sep-2024]].

Google Coral. (n.d.-d). Edge TPU Overview [[Online; accessed 29-Sep-2024]].

Google Coral. (n.d.-e). USB Accelerator [[Online; accessed 29-Sep-2024]].

Google Developers. (2021). Ml kit.

Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., & He, K. (2017). Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*.

Gpu delegate for tensorflow lite [Accessed: 2024-09-27]. (2024).

Graves, A., et al. (2013). Speech recognition with deep recurrent neural networks. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 6645–6649.

Gu, T., et al. (2017). Badnets: Identifying vulnerabilities in the machine learning model supply chain.

Gubbi, J., et al. (2013). Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7), 1645–1660.

Gulshan, V., Peng, L., Coram, M., Stumpe, M. C., Wu, D., Narayanaswamy, A., Webster, D. R., et al. (2016). Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *JAMA*, 316(22), 2402–2410. https://doi.org/10.1001/jama.2016.17216

Gunning, D. (2017). Explainable artificial intelligence (xai). *Defense Advanced Research Projects Agency (DARPA)*.

Gupta, R., & Tanwar, S. (2021). Trust management in edge computing: A blockchain-based approach. *IEEE Transactions on Industrial Informatics*, 17(2), 1238–1247. https://doi.org/10.1109/TII.2020.3003653

Gurman, M. (2023). Apple works on ai tools to challenge openai and google.

Haj-Ali, A., et al. (2019). A tensorflow frontend for high-performance and hardware-agnostic machine learning. *arXiv preprint arXiv:1905.08369*.

Halevi, S., & Shoup, V. (2014). Algorithms in helib. *Advances in Cryptology – CRYPTO 2014*, 554–571.

Halfhill, T. R. (2013). A new era for optical computing. *Microprocessor Report*, 27(9), 1–3.

Han, S., Mao, H., & Dally, W. J. (2016a). Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Han, S., et al. (2016b). Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *International Conference on Learning Representations*.

Han, S., et al. (2017). Ese: Efficient speech recognition engine with sparse lstm on fpga. *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 75–84.

Han, S., Mao, H., & Dally, W. J. (2015). Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*.

Han, S., Mao, H., & Dally, W. J. (2016c). Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *International Conference on Learning Representations (ICLR)*.

Han, S., Pool, J., Tran, J., & Dally, W. (2015). Learning both weights and connections for efficient neural networks. *Neural Information Processing Systems (NeurIPS)*.

Hao, J., et al. (2018). Towards efficient and privacy-preserving computing in big data era. *IEEE Transactions on Services Computing*, 11(1), 167–178.

Harris, N. C., et al. (2018). Linear programmable nanophotonic processors. *Optica*, 5(12), 1623–1631.

He, Y., Annavaram, M., & Avestimehr, S. (2019a). Group knowledge transfer: Federated learning of large cnns at the edge. *Advances in Neural Information Processing Systems*, 32, 14068–14080.

He, Y., et al. (2019b). Streaming end-to-end speech recognition for mobile devices. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 999–1003.

He, Y., Zhang, X., & Sun, J. (2017). Channel pruning for accelerating very deep neural networks. *International Conference on Computer Vision (ICCV)*.

Heafield, K., et al. (2016). Recurrent neural network grammar for speech recognition. *Interspeech*, 765–769.

Helbing, D. (2019). Societal, economic, ethical and legal challenges of the digital revolution: From big data to deep learning, artificial intelligence, and manipulative technologies. In *Towards digital enlightenment* (pp. 47–72). Springer.

Hendrycks, D., & Gimpel, K. (2017). A baseline for detecting misclassified and out-of-distribution examples in neural networks. *International Conference on Learning Representations*.

Hinton, G., Vinyals, O., & Dean, J. (2015a). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Hinton, G., Vinyals, O., & Dean, J. (2015b). Distilling the knowledge in a neural network. *Neural Information Processing Systems (NeurIPS) Workshops*.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.

Hong, M., & Gonzalez, J. E. (2020). Efficient neural network inference on edge devices. *IEEE Micro*, 40(5), 28–35.

Horowitz, M. (2014). Computing's energy problem (and what we can do about it). *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, 10–14.

Howard, A., et al. (2019a). Searching for mobilenetv3. *Proceedings of the IEEE International Conference on Computer Vision*, 1314–1324.

Howard, A. G., et al. (2017a). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., et al. (2017b). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*. https://doi.org/10.48550/arXiv.1704.04861

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., et al. (2019b). Searching for mobilenetv3. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Howard, D. (2018). Enabling efficient ml for edge devices with arm's project trillium.

Hoy, M. B. (2018a). Alexa, siri, cortana, and more: An introduction to voice assistants. *Medical Reference Services Quarterly*, 37(1), 81–88.

Hoy, M. B. (2018b). Alexa, siri, cortana, and more: An introduction to voice assistants. *Medical Reference Services Quarterly*, 37(1), 81–88.

Hoy, M. B. (2018c). Alexa, siri, cortana, and more: An introduction to voice assistants. *Medical Reference Services Quarterly*, 37(1), 81–88. https://doi.org/10.1080/02763869.2018.1404391

Hua, X., Liu, L., Yang, T., Zhao, N., & Sun, Z. (2020). Blockchain-based federated learning for intelligent control in heavy haul railway. *IEEE Access*, 8, 176830–176839.

Huang, X., & Li, J. (2020). Preventing free-riding in decentralized edge networks. *IEEE Network*, 34(2), 214–221. https://doi.org/10.1109/MNET.001.1900382

Huawei. (n.d.). Ascend AI Processor [[Online; accessed 29-Sep-2024]].

Huawei. (2019). Kirin 990 5g: World's first flagship 5g soc.

Huawei. (2021). Ascend ai processor series.

Hung, S., et al. (2016). *Mobile edge computing (mec): A key technology towards 5g* (tech. rep.). ETSI White Paper No. 11.

Iandola, F. N., et al. (2016). Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5mb model size. *arXiv preprint arXiv:1602.07360*.

Ielmini, D., & Wong, H.-S. P. (2018). In-memory computing with resistive switching devices. *Nature Electronics*, 1(6), 333–343.

Ignatov, A., et al. (2019). Ai benchmark: All about deep learning on smartphones in 2019. *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 3617–3635.

IMT-2030. (2020). *Framework and overall objectives of the future development of imt for 2030 and beyond* (tech. rep.). International Telecommunication Union.

Inc., A. (2020a). Iphone 12 pro and iphone 12 pro max: The most powerful iphones ever with advanced technologies.

Inc., A. (2020b). Iphone 12 pro and iphone 12 pro max: The most powerful iphones ever with advanced technologies [Accessed: 2024-09-20].

Inc., A. (2020c). Siri learning guide.

Inc., A. (2021a). Apple introduces on-device processing for siri requests. *Apple Newsroom*.

Inc., A. (2021b). Siri data and privacy overview.

Intel. (n.d.-a). Intel Movidius Myriad X VPU [[Online; accessed 29-Sep-2024]].

Intel. (n.d.-b). Intel Neural Compute Stick 2 [[Online; accessed 29-Sep-2024]].

Intel. (n.d.-c). OpenVINO Toolkit [[Online; accessed 29-Sep-2024]].

Intel. (n.d.-d). Supported Frameworks and Layers [[Online; accessed 29-Sep-2024]].

Intel. (2016). Data is the new oil in the future of automated driving.

International Energy Agency. (2021). Data centres and data transmission networks.

Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., et al. (2018a). Quantization and training of neural networks for efficient integer-arithmetic-only inference. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2704–2713. https://doi.org/10.1109/CVPR.2018.00286

Jacob, B., Kligys, S., Chen, B., et al. (2018b). Quantization and training of neural networks for efficient integer-arithmetic-only inference. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2704–2713. https://doi.org/10.1109/CVPR.2018.00286

Jain, A. K., Ross, A., & Nandakumar, K. (2011). *Introduction to biometrics*. Springer Science & Business Media.

Jain, A. K., et al. (2020). Quality control in manufacturing using ai and edge computing. *IEEE Embedded Systems Letters*, 12(3), 81–84.

Janssen, M., et al. (2019). Big and open linked data (bold) in government: A challenge to transparency and privacy? *Government Information Quarterly*, 29(1), 112–118.

Jiang, W., et al. (2020a). Accelerating deep learning inference with algorithm and hardware co-design. *ACM Transactions on Embedded Computing Systems*, 19(6), 1–23.

Jiang, W., et al. (2020b). Collaborative deep learning in edge computing for recognition of human activities. *IEEE Transactions on Industrial Informatics*, 16(3), 1973–1983.

Jiang, Z., Li, C., Ye, M., & Ma, Z. (2021). Cross-platform deep learning model deployment for edge devices. *IEEE Access*, 9, 79569–79580.

Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., & Wang, F. (2020). Tinybert: Distilling bert for natural language understanding. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 4163–4174. https://doi.org/10.18653/v1/2020.emnlp-main.346

Joachims, T. (2006). Training linear svms in linear time. *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 217–226.

Jocher, G., et al. (2020). Yolov5 nano: A small and fast object detection model [[Online; accessed 29-Sep-2024]].

Jolliffe, I. T. (2002). *Principal component analysis*. Springer.

Jones, N. (2018). How to stop data centres from gobbling up the world's electricity. *Nature*, 561(7722), 163–166. https://doi.org/10.1038/d41586-018-06610-y

Jouppi, N. P., Young, C., Patil, N., Patterson, D., Agrawal, G., Bajwa, R., et al. (2017a). In-datacenter performance analysis of a tensor processing unit. *Proceedings of the 44th Annual International Symposium on Computer Architecture*, 1–12. https://doi.org/10.1145/3079856.3080246

Jouppi, N. P., Young, C., Patil, N., Patterson, D., Agrawal, G., Bajwa, R., et al. (2017b). In-datacenter performance analysis of a tensor processing unit. *Proceedings of the 44th Annual International Symposium on Computer Architecture*, 1–12. https://doi.org/10.1145/3079856.3080246

Jouppi, N. P., et al. (2017c). In-datacenter performance analysis of a tensor processing unit. *Proceedings of the 44th Annual International Symposium on Computer Architecture*, 1–12.

Jouppi, N. P., et al. (2018). A domain-specific architecture for deep neural networks. *Communications of the ACM*, 61(9), 50–59.

Juan, D. C., et al. (2018). Hardware-software co-design for deep learning. *Design Automation Conference (DAC)*, 1–6.

Kaiser, J., et al. (2020). Synaptic plasticity dynamics for deep continuous local learning (decolle). *Frontiers in Neuroscience*, 14, 424.

Kanerva, P. (2009). Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors. *Cognitive Computation*, 1(2), 139–159.

Kang, J., Yu, R., Huang, X., Maharjan, S., Zhang, Y., & Hossain, E. (2018). Blockchain for secure and efficient data sharing in vehicular edge computing and networks. *IEEE Communications Magazine*, 56(8), 62–68. https://doi.org/10.1109/MCOM.2018.1700879

Kang, J., Yu, R., Huang, X., Maharjan, S., Zhang, Y., & Hossain, E. (2020). Reliable federated learning for mobile networks. *IEEE Wireless Communications*, 27(2), 72–80.

Kang, J., Yu, R., Huang, X., Maharjan, S., Zhang, Y., & Hossain, E. (2017). Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains. *IEEE Transactions on Industrial Informatics*, 13(6), 3154–3164.

Kang, Y., Hauswald, J., Gao, C., Rovinski, A., Mudge, T., Mars, J., & Tang, L. (2017). Neurosurgeon: Collaborative intelligence between the cloud and mobile edge. *ACM SIGARCH Computer Architecture News*, 45(1), 615–629. https://doi.org/10.1145/3093337.3037698

Kang, Y., Hauswald, J., Rovinski, A., Mudge, T., Mars, J., & Tang, L. (2017). Neurosurgeon: Collaborative intelligence between the cloud and mobile edge. *Proceedings of the 22nd International Conference on Architectural Support for Programming Languages and Operating Systems*, 615–629.

Kapania, N. R., & Gerdes, J. C. (2015). Designing steering feel for autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 16(5), 2442–2451.

Khatoun, R., & Zeadally, S. (2016). Smart cities: Concepts, architectures, research opportunities. *Communications of the ACM*, 59(8), 46–57.

Khemka, A. (2021). The future of ai is hybrid: Balancing edge and cloud.

Khosravi, H., & Cooper, K. (2018). Personalised learning analytics: An integrative approach to adaptivity. *Journal of Learning Analytics*, 5(1), 79–97.

Kim, H., & Kim, Y. (2021). Lending and borrowing computational resources in decentralized edge networks. *Sensors*, 21(3), 892.

Kim, M., et al. (2018). Secure multi-party computation for federated learning. *IEEE International Conference on Information Fusion*, 1–8.

Kim, M., Park, J., Bennis, M., & Kim, S.-L. (2019). On-device federated learning via blockchain and its latency analysis. *2019 IEEE International Conference on Communications (ICC)*, 1–7. https://doi.org/10.1109/ICC.2019.8761315

Kim, Y., et al. (2020). Dynamic layer scaling for neural machine translation. *arXiv preprint arXiv:2004.10069*.

Kim, Y. (2014). Convolutional neural networks for sentence classification. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 1746–1751.

Kim, Y., Park, E., & Yoo, S. (2016). Compression of deep convolutional neural networks for fast and low-power mobile applications. *International Conference on Learning Representations (ICLR)*.

Klei, M. (2017). Personal finance and technology: The digital wallet. *Journal of Financial Planning*, 30(4), 16–17.

Koeberl, S., et al. (2014). Trustlite: A security architecture for tiny embedded devices. *Proceedings of the 9th European Conference on Computer Systems*, 1–14.

Kokku, R., et al. (2012). Nvs: A substrate for virtualizing wireless resources in cellular networks. *IEEE/ACM Transactions on Networking*, 20(5), 1333–1346.

Konečný, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., & Bacon, D. (2016a). Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*. https://doi.org/10.48550/arXiv.1610.05492

Konečný, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., & Bacon, D. (2016b). Federated learning: Strategies for improving communication efficiency. https://doi.org/10.48550/arXiv.1610.05492

Konečný, J., et al. (2016c). Federated learning: Strategies for improving communication efficiency.

Konečný, J., et al. (2016d). Federated optimization: Distributed machine learning for on-device intelligence.

Krishnamoorthi, R. (2018). Quantizing deep convolutional networks for efficient inference: A whitepaper. *arXiv preprint arXiv:1806.08342*. https://doi.org/10.48550/arXiv.1806.08342

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 1097–1105.

Kudo, T., & Richardson, J. (2018). Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 66–71.

Kugler, L. (2018). The next frontier of ai: Edge computing. *Communications of the ACM*, 61(12), 15–16. https://doi.org/10.1145/3276744

Kumar, S., et al. (2017). Resource-efficient machine learning in 2 kb ram for the internet of things. *Advances in Neural Information Processing Systems*, 30, 1935–1945.

Kurakin, A., et al. (2017). Adversarial machine learning at scale. *International Conference on Learning Representations*.

Kwon, H., et al. (2020). Efficient neural network compression. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 10(4), 522–535.

Lan, Z. (2019). Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Lane, N. D., et al. (2016). Deepx: A software accelerator for low-power deep learning inference on mobile devices. *Proceedings of the 15th International Conference on Information Processing in Sensor Networks*, 1–12.

Lane, N. D., Bhattacharya, S., Georgiev, P., Forlivesi, C., & Kawsar, F. (2015). An early resource characterization of deep learning on wearables, smartphones

and internet-of-things devices. *Proceedings of the International Workshop on Internet of Things towards Applications*, 7–12.

Lane, N. D., & Warden, P. (2018). The deep (learning) transformation of mobile and embedded computing. *IEEE Computer*, 51(5), 12–16.

Latva-aho, M., & Leppänen, K. (2019). Key drivers and research challenges for 6g ubiquitous wireless intelligence. *6G Flagship, University of Oulu*.

Laughlin, S. B., & Sejnowski, T. J. (2003). Communication in neuronal networks. *Science*, 301(5641), 1870–1874.

Leake, D. B., et al. (2018). Autonomous machines with nvidia jetson platform. *IEEE Micro*, 38(1), 17–29.

Lebedev, M. A., & Nicolelis, M. A. L. (2017). Brain-machine interfaces: From basic science to neuroprostheses and neurorehabilitation. *Physiological Reviews*, 97(2), 767–837.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.

LeCun, Y., Denker, J., & Solla, S. (1990). Optimal brain damage. *Advances in Neural Information Processing Systems (NeurIPS)*.

Lee, E. K., & Lee, Y. C. (2019). Incentive mechanisms for edge computing resource sharing: Survey and research challenges. *Sensors*, 19(21), 4727. https://doi.org/10.3390/s19214727

Lee, K.-F. (2018). *Ai superpowers: China, silicon valley, and the new world order*. Houghton Mifflin Harcourt.

Leng, C., Dou, D., Li, H., Zhu, S., & Jin, R. (2018). Extremely low bit neural networks: Squeeze the last bit out with admm. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Leporini, B., & Paternò, F. (2008). Applying web usability criteria for vision-impaired users: Does it really improve task performance? *International Journal of Human-Computer Interaction*, 24(1), 17–47.

Li, B., Li, Z., & Liu, J. (2018a). Deep learning-based object detection on autonomous driving vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 19(11), 3594–3608. https://doi.org/10.1109/TITS.2018.2838576

Li, B., Li, Z., & Liu, T. (2018). Deep learning-based object detection on autonomous driving vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 19(11), 3594–3608.

Li, B., Li, Z., & Liu, J. (2018b). Deep learning-based object detection on autonomous driving vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 19(11), 3594–3608. https://doi.org/10.1109/TITS.2018.2838576

Li, F., Luo, B., & Liu, P. (2010). Secure information aggregation for smart grids using homomorphic encryption. *First IEEE International Conference on Smart Grid Communications*, 327–332.

Li, H., et al. (2019). Learning simple algorithms from data: An example with edge computing. *IEEE Internet of Things Journal*, 6(6), 9878–9888.

Li, H., Kadav, A., Durdanovic, I., Samet, H., & Graf, H. P. (2017). Pruning filters for efficient convnets. *International Conference on Learning Representations (ICLR)*.

Li, M., Andersen, D. G., Park, J. W., Smola, A. J., Ahmed, A., Josifovski, V., & Su, B. Y. (2014). Scaling distributed machine learning with the parameter server. *11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14)*, 583–598.

Li, S., et al. (2017). Smart city: The state of the art, prototypes, and future research. *IEEE Communications Magazine*, 55(12), 122–131.

Li, S., et al. (2020). Edge intelligence for internet of things in 5g era: Vision, enabling technologies, and applications. *IEEE Internet of Things Journal*, 7(8), 6722–6747.

Li, T., Li, Y., & Wang, J. (2021). Integrating defi into edge ai: Opportunities and challenges. *IEEE Internet of Things Journal*, 8(12), 9816–9825.

Li, T., & Ma, H. (2021). Token economics for edge ai: Concepts and case studies. *IEEE Communications Magazine*, 59(6), 90–96. https://doi.org/10.1109/MCOM.001.2100017

Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. (2020). Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3), 50–60.

Li, X., Chen, J., Zhang, Y., & Vasilakos, A. V. (2018). Secure cache aided content delivery in mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 17(2), 304–319. https://doi.org/10.1109/TMC.2017.2719813

Li, X., & Wang, H. (2020). Edge intelligence: The confluence of edge computing and artificial intelligence. *IEEE Internet of Things Journal*, 7(8), 7457–7469.

Li, Y., Deng, L., Hoi, S. C. H., & Chen, Y. (2019). Deep learning for natural language processing: Advantages and challenges. *National Science Review*, 6(4), 442–446.

Li, Y., & Liu, M. (2018a). *Mobile edge computing empowered smart homes*. Springer.

Li, Y., Zhan, Y., Ren, J., & Yang, F. (2020). Computation offloading for edge computing with access control in smart healthcare systems. *Future Generation Computer Systems*, 107, 667–676.

Li, Y., & Liu, M. (2018b). *Mobile edge computing empowered smart homes*. Springer.

Li, Y., & Liu, M. (2018c). Mobile edge computing empowered smart homes. *Springer*.

Li, Z., et al. (2018). Edge-oriented computing paradigms: A survey on architecture design and system management. *ACM Computing Surveys*, 51(2), 1–34.

Liang, J., & Lee, J. (2022). Integrating large language models with robotics: A survey. *IEEE Transactions on Robotics*, 38(5), 2393–2408.

Liao, Q. V., et al. (2020). Questioning the ai: Informing design practices for explainable ai user experiences. *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 1–15.

Lichtsteiner, P., et al. (2008). A 128×128 120 db 15 s latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits*, 43(2), 566–576.

Lim, K., et al. (2012). Processor networking with 3d-stacked memory. *IEEE Micro*, 32(5), 22–31.

Lin, J., Yu, W., Zhang, N., Yang, X., Zhang, H., & Zhao, W. (2017a). A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications. *IEEE Internet of Things Journal*, 4(5), 1125–1142. https://doi.org/10.1109/JIOT.2017.2683200

Lin, J., Yu, W., Zhang, N., Yang, X., Zhang, H., & Zhao, W. (2017b). A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications. *IEEE Internet of Things Journal*, 4(5), 1125–1142. https://doi.org/10.1109/JIOT.2017.2683200

Lin, J. K., et al. (2020). Energy-efficient neural network accelerators: From cloud to edge. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 67(3), 615–624.

Lin, P., Abney, K., & Bekey, G. A. (2011). Robot ethics: Mapping the issues for a mechanized world. *Artificial Intelligence*, 175(5-6), 942–949.

Lin, W., & Wang, H. (2019). Scalability challenges in edge ai: A survey. *IEEE Transactions on Industrial Informatics*, 15(7), 4239–4247. https://doi.org/10.1109/TII.2019.2901271

Lin, Y., et al. (2018). Energy-efficient asr for embedded devices. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5469–5473.

Litman, T. (2019). Autonomous vehicle implementation predictions. *Victoria Transport Policy Institute*, 28(1), 1–33.

Liu, A., et al. (2017). Secure and privacy preserving data aggregation scheme for fog computing-based smart grids. *IEEE Access*, 5, 5326–5339.

Liu, H., Simonyan, K., & Yang, Y. (2019). Darts: Differentiable architecture search. *International Conference on Learning Representations (ICLR)*.

Liu, H., et al. (2000). Adaptive neural network control of robot manipulators with uncertain kinematics and dynamics. *IEEE Transactions on Automatic Control*, 45(1), 176–181.

Liu, J., Tang, J., Xu, Y., & Zhang, W. (2020). Computation offloading and content caching in wireless cellular networks with mobile edge computing. *IEEE Transactions on Vehicular Technology*, 69(2), 2285–2299.

Liu, X., et al. (2018). Trojaning attack on neural networks. *Network and Distributed System Security Symposium*.

Liu, Y., Ning, P., & Li, Y. (2014). *Handbook of software and hardware trojan detection*. CRC Press.

Liu, Y., Peng, K., Ning, Z., Wang, H., Guo, L., & Guo, S. (2020). Resource allocation in autonomous driving networks: A joint computation, caching, and communication perspective. *IEEE Transactions on Intelligent Transportation Systems*, 21(11), 4793–4804.

Liu, Y., & Zhang, X. (2019). Staking mechanisms in blockchain networks: A survey. *Journal of Blockchain Research*, 2(1), 45–59.

Liu, Z., & Li, X. (2019). Reputation systems and penalties in defi-based edge computing. *ACM Transactions on Internet Technology*, 19(4), 51.

Louis, P., et al. (2019). Protecting neural networks with model steganography. *Advances in Neural Information Processing Systems*, 32, 1536–1546.

Ltd., A. (2021). Arm trustzone technology.

Lu, X., & Li, J. (2020a). Speed is all you need: On-device acceleration of large-scale conversational ai. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(9), 13523–13530.

Lu, X., & Li, J. (2020b). Speed is all you need: On-device acceleration of large-scale conversational ai. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(09), 13523–13530.

Lu, Y. (2019). Blockchain and federated learning for collaborative intrusion detection in iot: A survey. *Wireless Communications and Mobile Computing*, 2019, 1–10. https://doi.org/10.1155/2019/1037595

Lu, Y., et al. (2021a). Blockchain and federated learning for collaborative intrusion detection in edge computing. *IEEE Transactions on Industrial Informatics*, 17(7), 4962–4970.

Lu, Y., Liu, C., Wang, K. I.-K., Huang, H., & Xu, X. (2020). Digital twin-driven smart manufacturing: Connotation, reference model, applications and research issues. *Robotics and Computer-Integrated Manufacturing*, 61, 101837. https://doi.org/10.1016/j.rcim.2019.101837

Lu, Y., et al. (2021b). Blockchain and federated learning for collaborative intrusion detection in edge computing. *IEEE Transactions on Industrial Informatics*, 17(7), 4962–4970.

Lubart, T. (2005). How can computers be partners in the creative process: Classification and commentary on the special issue. *International Journal of Human-Computer Studies*, 63(4-5), 365–369.

Lyu, L., Yu, H., & Yang, Q. (2020). Threats to federated learning: A survey. *arXiv preprint arXiv:2003.02133*.

Ma, X., Ding, Y., Wang, X., & Wang, J. (2018a). Cloud-assisted privacy-preserving mobile health monitoring. *IEEE Access*, 6, 36552–36561.

Ma, X., Ding, Y., Wang, X., & Wang, J. (2018b). Cloud-assisted privacy-preserving mobile health monitoring. *IEEE Access*, 6, 36552–36561.

Maass, W. (1997). Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, 10(9), 1659–1671.

Mach, P., & Becvar, Z. (2017a). Mobile edge computing: A survey on architecture and computation offloading. *IEEE Communications Surveys & Tutorials*, 19(3), 1628–1656. https://doi.org/10.1109/COMST.2017.2682318

Mach, P., & Becvar, Z. (2017b). Mobile edge computing: A survey on architecture and computation offloading. *IEEE Communications Surveys & Tutorials*, 19(3), 1628–1656. https://doi.org/10.1109/COMST.2017.2682318

Mach, P., & Becvar, Z. (2017c). Mobile edge computing: A survey on architecture and computation offloading. *IEEE Communications Surveys & Tutorials*, 19(3), 1628–1656. https://doi.org/10.1109/COMST.2017.2682318

Madry, A., et al. (2018). Towards deep learning models resistant to adversarial attacks. *International Conference on Learning Representations*.

Maes, P. (1994). Agents that reduce work and information overload. *Communications of the ACM*, 37(7), 30–40.

Mahmoud, M. S., & Mohamad, M. S. (2019). A study of efficient power consumption wireless communication techniques/modules for internet of things (iot) applications. *Advances in Internet of Things*, 9(2), 19–29.

Mann, S. (2014). Wearable computing. In *Encyclopedia of human-computer interaction* (2nd).

Mao, Y., You, C., Zhang, J., Huang, K., & Letaief, K. B. (2017). A survey on mobile edge computing: The communication perspective. *IEEE Communications Surveys & Tutorials*, 19(4), 2322–2358.

Mao, Y., Zhang, J., & Letaief, K. B. (2017). Mobile edge computing: Energy-efficient offloading of mobile computing to edge clouds with computing latency constraints. *IEEE Transactions on Wireless Communications*, 16(7), 4809–4822. https://doi.org/10.1109/TWC.2017.2695585

Mao, Y., et al. (2017). A survey on mobile edge computing: The communication perspective. *IEEE Communications Surveys & Tutorials*, 19(4), 2322–2358.

Markram, H., et al. (1997). Regulation of synaptic efficacy by coincidence of postsynaptic aps and epsps. *Science*, 275(5297), 213–215.

Marston, S., Li, Z., Bandyopadhyay, S., Zhang, J., & Ghalsasi, A. (2011). Cloud computing—the business perspective. *Decision Support Systems*, 51(1), 176–189.

Martin, K. E. (2019). Ethical issues in the big data industry. *MIS Quarterly Executive*, 18(2), 209–232.

McDuff, D., & Czerwinski, M. (2018). Designing emotionally sentient agents. *Communications of the ACM*, 61(12), 74–83.

McMahan, B., et al. (2017). Communication-efficient learning of deep networks from decentralized data. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 1273–1282.

McMahan, B., et al. (2016). Federated learning of deep networks using model averaging. *arXiv preprint arXiv:1602.05629*.

McMahan, B., & Ramage, D. (2017). Federated learning: Collaborative machine learning without centralized training data.

McMahan, H. B., et al. (2016). Federated averaging: A simple and robust federated learning algorithm.

Mead, C. (1990). Neuromorphic electronic systems. *Proceedings of the IEEE*, 78(10), 1629–1636.

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2021). A survey on bias and fairness in machine learning. *ACM Computing Surveys*, 54(6), 1–35.

Meiklejohn, S., & Mercer, R. (2018). Möbius: Trustless tumbling for transaction privacy. *Proceedings on Privacy Enhancing Technologies*, 2018(2), 105–121.

Merolla, P. A., et al. (2014). A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345(6197), 668–673.

Meta AI Research. (2023). Llama: Open and efficient foundation language models.

Microsoft. (2021). Translator app features.

Microsoft. (2024). Onnx runtime: Cross-platform, high performance scoring engine for open neural network exchange (onnx) models. https://onnxruntime.ai/

Microsoft Azure. (2021a). Azure iot edge.

Microsoft Azure. (2021b). Azure iot edge [Retrieved from https://azure.microsoft.com/services/iot-edge/].

Miers, I., Garman, C., Green, M., & Rubin, A. D. (2013). Zerocoin: Anonymous distributed e-cash from bitcoin. *IEEE Symposium on Security and Privacy*, 397–411.

Miller, D. A. B. (2017). Attojoule optoelectronics for low-energy information processing and communications. *Journal of Lightwave Technology*, 35(3), 346–396.

Mishra, N., et al. (2008). Context-aware energy enhancement in sensor nodes. *Proceedings of the IEEE International Conference on Distributed Computing in Sensor Systems*, 1–10.

Mittal, S. (2019). A survey on optimized implementation of deep learning models on the nvidia jetson platform. *Journal of Systems Architecture*, 97, 428–442.

Molchanov, P., Tyree, S., Karras, T., Aila, T., & Kautz, J. (2017). Pruning convolutional neural networks for resource efficient inference. *International Conference on Learning Representations*. https://arxiv.org/abs/1611.06440.

Moloney, D. (2014). Myriad 2: Eye of the computational vision storm. *Hot Chips Symposium (HCS)*.

Moons, B., & Verhelst, M. (2016). An energy-efficient precision-scalable convolutional neural network accelerator. *Proceedings of the IEEE Symposium on VLSI Circuits*, 1–2.

Moore, G. E. (1965). Cramming more components onto integrated circuits. *Electronics*, 38(8).

Murshed, M. G., Murphy, C., Hou, D., Khan, M. A., Ananthanarayanan, G., & Zou, J. (2019). Machine learning at the network edge: A survey. *IEEE Internet of Things Journal*, 7(5), 4329–4346.

Nagel, M., Amjad, R., van Baalen, M., & Blankevoort, T. (2020). Up or down? adaptive rounding for post-training quantization. *International Conference on Machine Learning (ICML)*.

Nah, F. F.-H. (2004). A study on tolerable waiting time: How long are web users willing to wait? *Behaviour & Information Technology*, 23(3), 153–163. https://doi.org/10.1080/01449290410001669914

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system.

Narayanan, A., et al. (2019). Privacy-preserving machine learning. *Foundations and Trends in Privacy and Security*, 2(3), 151–157.

Nguyen, D. C., Ding, M., Pathirana, P. N., & Seneviratne, A. (2020). Blockchain and ai-based solutions for decentralized edge computing: A defi perspective. *IEEE Wireless Communications*, 27(6), 140–146.

Nguyen, D. C., Ding, M., Pathirana, P. N., & Seneviratne, A. (2021). Blockchain and ai-based solutions to combat coronavirus (covid-19)-like epidemics: A

survey. *IEEE Access*, 9, 95730–95753. https://doi.org/10.1109/ACCESS.2021.3095580

Nicolas-Alonso, L. F., & Gomez-Gil, J. (2012). Brain computer interfaces, a review. *Sensors*, 12(2), 1211–1279.

Nielsen, J. (1994). *Usability engineering*. Morgan Kaufmann.

Nnapi delegate for tensorflow lite [Accessed: 2024-09-27]. (2024).

Novikov, A., Podoprikhin, D., Osokin, A., & Vetrov, D. (2015). Tensorizing neural networks. *Advances in Neural Information Processing Systems (NeurIPS)*.

NVIDIA. (n.d.-a). CUDA Toolkit [[Online; accessed 29-Sep-2024]].

NVIDIA. (n.d.-b). Deep Learning Frameworks Support [[Online; accessed 29-Sep-2024]].

NVIDIA. (n.d.-c). JetPack SDK [[Online; accessed 29-Sep-2024]].

NVIDIA. (n.d.-d). Jetson Nano Developer Kit [[Online; accessed 29-Sep-2024]].

NVIDIA. (n.d.-e). NVDLA Deep Learning Accelerator [[Online; accessed 29-Sep-2024]].

NVIDIA. (n.d.-f). NVIDIA Jetson Nano GPU Architecture [[Online; accessed 29-Sep-2024]].

NVIDIA. (n.d.-g). Open Sourcing NVIDIA Deep Learning Accelerator [[Online; accessed 29-Sep-2024]].

NVIDIA. (n.d.-h). TensorRT [[Online; accessed 29-Sep-2024]].

NVIDIA. (2024). Tensorrt | nvidia developer [Online]. https://developer.nvidia.com/tensorrt

NVIDIA Corporation. (2019). Nvidia jetson nano developer kit.

NVIDIA Corporation. (2021a). Nvidia ai-on-5g platform.

NVIDIA Corporation. (2021b). Nvidia isaac platform for robotics.

NVIDIA Corporation. (2021c). Nvidia isaac platform for robotics [Retrieved from https://developer.nvidia.com/isaac-sdk].

NVIDIA Corporation. (2023). Nvidia jetson platform.

Oblinsky, A., et al. (2020). Model watermarking for recurrent neural networks.

of Health & Human Services, U. D. (2013a). Summary of the hipaa privacy rule.

of Health & Human Services, U. D. (2013b). Summary of the hipaa privacy rule [Accessed: 2024-09-20].

of Health & Human Services, U. D. (2021). Health information privacy.

Omoniwa, B., Hussain, R., Javed, M. A., Bouk, S. H., & Han, K. (2018a). Fog/edge computing-based iot (feciot): Architecture, applications, and research issues. *IEEE Internet of Things Journal*, 6(3), 4118–4149. https://doi.org/10.1109/JIOT.2018.2875544

Omoniwa, B., Hussain, R., Javed, M. A., Bouk, S. H., & Han, K. (2018b). Fog/edge computing-based iot (feciot): Architecture, applications, and research issues. *IEEE Internet of Things Journal*, 6(3), 4118–4149.

ONNX. (2024). Open neural network exchange (onnx). https://onnx.ai/

ONNX Runtime. (2023). Onnx runtime: Cross-platform, high performance ml inferencing and training accelerator.

OpenAI. (2022). Chatgpt: Optimizing language models for dialogue.

OpenAI. (2023). Gpt-4 technical report. https://doi.org/10.48550/arXiv.2303.08774

Openvino toolkit [Accessed: 2024-09-27]. (2024).

Optimizing tensorflow models for mobile and edge devices [Accessed: 2024-09-27]. (2024).

Paillier, P. (1999). Public-key cryptosystems based on composite degree residuosity classes. *Advances in Cryptology — EUROCRYPT '99*, 223–238.

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10), 1345–1359.

Papernot, N., et al. (2016). The limitations of deep learning in adversarial settings. *2016 IEEE European Symposium on Security and Privacy (EuroSP)*, 372–387.

Pardo, A., & Siemens, G. (2014). Ethical and privacy principles for learning analytics. *British Journal of Educational Technology*, 45(3), 438–450.

Parliament, E., & of European Union, C. (2016). Regulation (eu) 2016/679 (general data protection regulation).

Parzen, E., et al. (2017). User privacy and data protection in big data: A systematic literature review. *2017 IEEE International Conference on Big Data (Big Data)*, 2857–2866.

Pasquale, F. (2015). *The black box society: The secret algorithms that control money and information*. Harvard University Press.

Patel, S., Park, H., Bonato, P., Chan, L., & Rodgers, M. (2012). A review of wearable sensors and systems with application in rehabilitation. *Journal of NeuroEngineering and Rehabilitation*, 9(1), 21.

Patel, V., & Shah, M. (2021). Legal and privacy considerations in defi-based edge ai networks. *Journal of Information Security and Applications*, 58, 102717.

Patterson, D., Gonzalez, J., Le, Q., Liang, C., Munguia, L. M., Rothchild, D., et al. (2021a). Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350*. https://doi.org/10.48550/arXiv.2104.10350

Patterson, D., Gonzalez, J., Le, Q., Liang, C., Munguia, L. M., Rothchild, D., et al. (2021b). Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350*. https://doi.org/10.48550/arXiv.2104.10350

Paulin, M., Seldin, Y., et al. (2014). Transformation pursuit for semi-supervised clustering. *Proceedings of the NeurIPS*.

Peng, K., Zhang, Y., Wang, C., Qiao, X., Xu, Y., & Zhang, W. (2018). A survey on mobile edge computing: Focusing on service adoption and provision. *IEEE Access*, 6, 58249–58263. https://doi.org/10.1109/ACCESS.2018.2875681

Pérez, L., et al. (2021). Privacy-preserving federated learning: A blockchain and mpc-based solution. *arXiv preprint arXiv:2101.11248*.

Phan, A.-H., Nguyen, T. D., Lam, T. A., Abdel-Nasser, M., Ha, Q. K., Park, D., & Kim, D.-K. (2016). Robust bayesian low-rank factorization for image denoising. *International Conference on Image Processing (ICIP)*.

Picard, R. W. (2003). Affective computing: Challenges. *International Journal of Human-Computer Studies*, 59(1-2), 55–64.

Piwek, L., Ellis, D. A., Andrews, S., & Joinson, A. (2016). The rise of consumer health wearables: Promises and barriers. *PLOS Medicine*, 13(2), e1001953.

Plate, T. (1995). *Holographic reduced representation: Distributed representation for cognitive structures* [Doctoral dissertation, Stanford University].

Ponulak, F., & Kasinski, A. (2011). Introduction to spiking neural networks: Information processing, learning and applications. *Acta Neurobiologiae Experimentalis*, 71(4), 409–433.

Porambage, P., Okwuibe, J., Liyanage, M., Ylianttila, M., & Taleb, T. (2018). Survey on multi-access edge computing for internet of things realization. *IEEE Communications Surveys & Tutorials*, 20(4), 2961–2991. https://doi.org/10.1109/COMST.2018.2849509

Pouchet, L.-N., Singh, S., et al. (2017). Enabling efficient and scalable fpga-based acceleration of dnns. *Proceedings of the ACM SIGPLAN Symposium on Field-Programmable Gate Arrays (FPGA)*.

Pre-trained models for tensorflow lite [Accessed: 2024-09-27]. (2024).

Premsankar, G., et al. (2018a). Edge computing for the internet of things: A case study. *IEEE Internet of Things Journal*, 5(2), 1275–1284.

Premsankar, G., Di Francesco, M., & Taleb, T. (2018b). Edge computing for the internet of things: A case study. *IEEE Internet of Things Journal*, 5(2), 1275–1284. https://doi.org/10.1109/JIOT.2018.2805263

Puggelli, A., et al. (2018). A fully open-source isa to asic flow based on the risc-v architecture. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 37(1), 72–84.

PyTorch. (2020). Pytorch mobile.

PyTorch. (2024). Quantization support in pytorch [Online]. https://pytorch.org/docs/stable/quantization.html

PyTorch Mobile. (2023). Pytorch mobile | pytorch.

Qin, J., Li, W., Li, W., & Yu, J. (2020). A privacy-preserving mobile payment protocol based on blockchain. *IEEE Access*, 8, 181718–181727.

Qualcomm. (n.d.). Hexagon DSP [[Online; accessed 29-Sep-2024]].

Qualcomm Technologies, Inc. (2021a). The hybrid ai approach: Empowering devices with on-device and cloud ai.

Qualcomm Technologies, Inc. (2021b). The hybrid ai approach: Empowering devices with on-device and cloud ai [Retrieved from https://www.qualcomm.com/media/documents/files/hybrid-ai-whitepaper.pdf].

Qualcomm Technologies, Inc. (2023). Snapdragon mobile platforms: Bringing ai to the edge.

Rahimi, A., et al. (2016). Hyperdimensional computing for noninvasive brain–computer interfaces: Blind and one-shot classification of eeg error-related potentials. *International Conference on Rebooting Computing (ICRC)*, 1–8.

Rajendran, J., Rosenfeld, K., Tehranipoor, M., & Karri, R. (2012). Security analysis of integrated circuit camouflaging. *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, 709–720. https://doi.org/10.1145/2382196.2382279

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Sutskever, I., et al. (2021). Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092*. https://doi.org/10.48550/arXiv.2102.12092

Raspberry Pi Foundation. (n.d.-a). GPIO Pins [[Online; accessed 29-Sep-2024]].

Raspberry Pi Foundation. (n.d.-b). Raspberry Pi [[Online; accessed 29-Sep-2024]].

Raspberry Pi Foundation. (n.d.-c). Raspberry Pi 4 Model B [[Online; accessed 29-Sep-2024]].

Rastegari, M., Ordonez, V., Redmon, J., & Farhadi, A. (2016). Xnor-net: Imagenet classification using binary convolutional neural networks. *European Conference on Computer Vision (ECCV)*.

Rawat, D. B., et al. (2015). Cyber-physical systems and the internet of things: A survey. *Proceedings of the 11th International Conference on Ubiquitous Computing*, 1–9.

Reddy, M., & Chandra, S. (2020). User preference-based personalization in autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 21(5), 2092–2101.

Ren, J., et al. (2020). Accelerating edge intelligence through micro parallel computing: A hardware prototype perspective. *IEEE Wireless Communications*, 27(3), 82–88.

Ren, J., Zhang, D., He, S., Zhang, Y., & Li, T. (2019). A survey on end-edge-cloud orchestrated network computing paradigms: Transparent computing, mobile edge computing, fog computing, and cloudlet. *ACM Computing Surveys*, 52(6), 1–36. https://doi.org/10.1145/3362031

Ren, M., et al. (2021). Beyond model extraction: Inferring model hyperparameters using doubly black-box attacks. *International Conference on Learning Representations*.

Ribeiro, M. T., et al. (2016). Why should i trust you? explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135–1144.

Rieke, N., Hancox, J., Li, W., Milletari, F., Roth, H. R., Albarqouni, S., et al. (2020). The future of digital health with federated learning. *npj Digital Medicine*, 3(1), 1–7.

Rieke, S., et al. (2020). The future of digital health with federated learning. *NPJ Digital Medicine*, 3(119).

Ríos, C., et al. (2019). In-memory computing on a photonic platform. *Science Advances*, 5(2), eaau5759.

RISC-V Foundation. (n.d.). RISC-V: The Free and Open RISC Instruction Set Architecture [[Online; accessed 29-Sep-2024]].

Risko, E. F., & Gilbert, S. J. (2016). Cognitive offloading. *Trends in Cognitive Sciences*, 20(9), 676–688.

Ritchie, J., & Thomas, M. (2015). Ai for game developers. In *Ai game programming wisdom* (pp. 509–518). Charles River Media.

Romero, A., Ballas, N., Kahou, S., Chassang, A., Gatta, C., & Bengio, Y. (2015). Fitnets: Hints for thin deep nets. *International Conference on Learning Representations (ICLR)*.

Rosebrock, A. (2019). *Raspberry pi for computer vision*. PyImageSearch.

Rosenfeld, D., et al. (2010). Flood or drought: How do aerosols affect precipitation? *Science*, 321(5894), 1309–1313.

Rudenko, A., et al. (1998). Saving portable computer battery power through remote process execution. *ACM SIGMOBILE Mobile Computing and Communications Review*, 2(1), 19–26.

Rueckauer, B., et al. (2017). Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Frontiers in Neuroscience*, 11, 682.

Rührmair, U., et al. (2010). Security applications of pufs. *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 1–6.

Sabt, M., Achemlal, M., & Bouabdallah, A. (2015). Trusted execution environment: What it is, and what it is not. *2015 IEEE Trustcom/BigDataSE/ISPA*, 57–64. https://doi.org/10.1109/Trustcom.2015.357

Sahai, A., & Waters, B. (2014). How to use indistinguishability obfuscation: Deniable encryption, and more. *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, 475–484.

Sainath, T. N., Kingsbury, B., Sindhwani, V., Arisoy, E., & Ramabhadran, B. (2013). Low-rank matrix factorization for deep neural network training with high-dimensional output targets. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

Sajnani, R., & Thilakarathna, K. (2020). Decentralized edge intelligence: A dynamic resource management framework for the edge-cloud continuum. *IEEE Transactions on Mobile Computing*, 19(10), 2305–2322.

Sallouha, H., et al. (2017). Localization in long-range ultra narrow band iot networks using rssi. *Proceedings of the IEEE International Conference on Communications (ICC)*, 1–6.

Samsung Electronics. (2021). Exynos processors with advanced ai capabilities.

Sandler, M., et al. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4510–4520.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019a). Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Sanh, V., et al. (2019b). Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Satyanarayanan, M. (2017a). The emergence of edge computing. *Computer*, 50(1), 30–39.

Satyanarayanan, M. (2017b). The emergence of edge computing. *Computer*, 50(1), 30–39. https://doi.org/10.1109/MC.2017.9

Satyanarayanan, M. (2017c). The emergence of edge computing. *Computer*, 50(1), 30–39. https://doi.org/10.1109/MC.2017.9

Satyanarayanan, M. (2017d). The emergence of edge computing. *Computer*, 50(1), 30–39.

Schär, F. (2021). Decentralized finance: On blockchain- and smart contract-based financial markets. *Federal Reserve Bank of St. Louis Review*, 103(2), 153–174.

Schuld, M., Sinayskiy, I., & Petruccione, F. (2015). An introduction to quantum machine learning. *Contemporary Physics*, 56(2), 172–185.

Schulz, K., & Mayer, H. (2018). Short-term wind and solar power forecasts: An overview. *Renewable and Sustainable Energy Reviews*, 14(7), 1543–1561.

Schuster, M., Paliwal, K. K., & Sim, K. C. (2020a). On-device end-to-end speech recognition. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing*, 6059–6063. https://doi.org/10.1109/ICASSP40776.2020.9054251

Schuster, M., Paliwal, K. K., & Sim, K. C. (2020b). On-device end-to-end speech recognition. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing*, 6059–6063. https://doi.org/10.1109/ICASSP40776.2020.9054251

Sebastian, A., et al. (2019). Rapid and efficient object recognition using ultra-efficient hyperdimensional computing. *Nature Electronics*, 2(12), 521–529.

Sengupta, A., et al. (2020). Spintronics for probabilistic computing and learning. *Nature Electronics*, 3(6), 363–376.

Seo, D., et al. (2016). Neural dust: An ultrasonic, low power solution for chronic brain-machine interfaces. *arXiv preprint arXiv:1605.06287*.

Seo, J.-S., et al. (2011). A 45nm cmos neuromorphic chip with a scalable architecture for learning in networks of spiking neurons. *2011 IEEE Custom Integrated Circuits Conference (CICC)*, 1–4.

Services, A. W. (2021). Aws machine learning. https://aws.amazon.com/machine-learning/

Shamir, A. (1979). How to share a secret. *Communications of the ACM*, 22(11), 612–613.

Sharif, M., et al. (2016). Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 1528–1540.

Shayan, M., Fung, C., Mohammadi, M., & Ngai, E. C.-H. (2020). Biscotti: A blockchain system for private and secure federated learning. *IEEE Transactions on Parallel and Distributed Systems*, 32(7), 1513–1525.

Shearer, E., & Gottfried, J. (2017). News use across social media platforms 2017.

Shen, Y., et al. (2017). Deep learning with coherent nanophotonic circuits. *Nature Photonics*, 11(7), 441–446.

Shen, Y., et al. (2018). An integrated-nanophotonics accelerator for neural networks. *Optics Express*, 26(6), 7313–7331.

Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016a). Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5), 637–646. https://doi.org/10.1109/JIOT.2016.2579198

Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016b). Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5), 637–646. https://doi.org/10.1109/JIOT.2016.2579198

Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016c). Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5), 637–646. https://doi.org/10.1109/JIOT.2016.2579198

Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016d). Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5), 637–646.

Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016e). Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5), 637–646.

Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016f). Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5), 637–646. https://doi.org/10.1109/JIOT.2016.2579198

Shi, W., & Dustdar, S. (2016). The promise of edge computing. *Computer*, 49(5), 78–81.

Shokri, R., et al. (2017). Membership inference attacks against machine learning models. *2017 IEEE Symposium on Security and Privacy (SP)*, 3–18.

Shokri, R., & Shmatikov, V. (2015). Privacy-preserving deep learning. *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 1310–1321.

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*.

Siau, K., & Wang, W. (2018). Building trust in artificial intelligence, machine learning, and robotics. *ACM Transactions on Management Information Systems*, 9(3), 7.

SiFive. (n.d.). SiFive Intelligence Processors [[Online; accessed 29-Sep-2024]].

Silver, D., et al. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587), 484–489.

Simonyan, A., et al. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications.

Singh, A., & Chatterjee, S. (2020). Resource liquidity pools for edge ai applications using defi. *Future Internet*, 12(10), 168.

Singh, A., & Sharma, N. (2019). Edge computing in autonomous vehicles: Opportunities and challenges. *IEEE Internet of Things Magazine*, 2(1), 26–31. https://doi.org/10.1109/IOTM.0001.1900013

Singhal, K., Azizi, S., Tu, T., Mahdavi, S. S., Wei, J., Chung, H. W., et al. (2022). Large language models encode clinical knowledge. *arXiv preprint arXiv:2212.13138*. https://doi.org/10.48550/arXiv.2212.13138

Smith, T., et al. (2017). Federated multi-task learning. *Advances in Neural Information Processing Systems*, 30, 4424–4434.

Song, D., et al. (2019). Differential privacy via compressing gradients. *Advances in Neural Information Processing Systems*, 32, 3700–3710.

Song, J., et al. (2018). Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. *International Conference on Learning Representations*.

Sood, S. K., & Mahajan, I. (2017). Wearable iot sensor based healthcare system for identifying and controlling chikungunya virus. *Computers in Industry*, 91, 33–44. https://doi.org/10.1016/j.compind.2017.05.003

Stojkoska, B. L. R., & Trivodaliev, K. V. (2017). A review of internet of things for smart home: Challenges and solutions. *Journal of Cleaner Production*, 140, 1454–1464.

Stromatias, E., et al. (2015). Robustness of spiking deep belief networks to noise and reduced bit precision of neuro-inspired hardware platforms. *Frontiers in Neuroscience*, 9, 222.

Strommer, T., et al. (2020). Tinyml: Machine learning for resource-constrained environments. *Proceedings of the NeurIPS Workshops*.

Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and policy considerations for deep learning in nlp. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3645–3650. https://doi.org/10.18653/v1/P19-1355

Sun, S., et al. (2020). Mobilebert: A compact task-agnostic bert for resource-limited devices. *arXiv preprint arXiv:2004.02984*.

Sutskever, I., et al. (2014). Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems*, 27, 3104–3112.

Systems, C. (2018a). Cisco visual networking index: Forecast and trends, 2017–2022.

Systems, C. (2018b). Cisco visual networking index: Forecast and trends, 2017–2022 [Accessed: 2024-09-20].

Szabo, N. (1997). Formalizing and securing relationships on public networks. *First Monday*, 2(9). https://doi.org/10.5210/fm.v2i9.548

Sze, V., Chen, Y. H., Yang, T. J., & Emer, J. S. (2017a). Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, 105(12), 2295–2329. https://doi.org/10.1109/JPROC.2017.2761740

Sze, V., Chen, Y. H., Yang, T. J., & Emer, J. S. (2017b). Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, 105(12), 2295–2329. https://doi.org/10.1109/JPROC.2017.2761740

Sze, V., Chen, Y.-H., Yang, T.-J., & Emer, J. S. (2017). Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*.

Szegedy, C., et al. (2013). Intriguing properties of neural networks.

Tait, A. N., et al. (2017). Neuromorphic photonic networks using silicon photonic weight banks. *Scientific Reports*, 7(1), 7430.

Takabi, H., Joshi, J. B., & Ahn, G. J. (2010). Security and privacy challenges in cloud computing environments. *IEEE Security & Privacy*, 8(6), 24–31.

Taleb, T., Samdanis, K., Mada, B., Flinck, H., Dutta, S., & Sabella, D. (2017a). On multi-access edge computing: A survey of the emerging 5g network edge cloud architecture and orchestration. *IEEE Communications Surveys & Tutorials*, 19(3), 1657–1681. https://doi.org/10.1109/COMST.2017.2705720

Taleb, T., Samdanis, K., Mada, B., Flinck, H., Dutta, S., & Sabella, D. (2017b). On multi-access edge computing: A survey of the emerging 5g network edge cloud architecture and orchestration. *IEEE Communications Surveys & Tutorials*, 19(3), 1657–1681.

Tan, H., Pan, S., Li, Y., & Wu, Z. (2018). A survey of deep learning-based distributed training optimization. *IEEE Access*, 7, 142331–142346.

Tan, M., & Le, Q. V. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. *Proceedings of the 36th International Conference on Machine Learning*, 6105–6114.

Tan, M., & Le, Q. V. (2021). Efficientnetv2: Smaller models and faster training. *Proceedings of the 38th International Conference on Machine Learning*, 10096–10106.

Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., & Le, Q. V. (2019). Mnasnet: Platform-aware neural architecture search for mobile. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tang, J., et al. (2016). Ensuring security and privacy preservation for cloud data services. *ACM Computing Surveys*, 49(1), 1–39.

Tang, Z., et al. (2020). Quantized neural networks for low-power embedded systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 39(1), 142–151.

Tao, F., et al. (2019). Digital twins and cyber–physical systems toward smart manufacturing and industry 4.0: Correlation and comparison. *Engineering*, 5(4), 653–661.

Taylor, L., & Nitschke, P. (2018). Data augmentation methods for deep learning. *International Conference on Computer Vision Workshops (ICCV)*.

TensorFlow. (n.d.-a). TensorFlow Lite on Raspberry Pi [[Online; accessed 29-Sep-2024]].

TensorFlow. (n.d.-b). TensorFlow Models on Edge TPU [[Online; accessed 29-Sep-2024]].

TensorFlow. (2021a). Tensorflow lite.

TensorFlow. (2021b). Tensorflow lite.

TensorFlow. (2021c). Tensorflow model optimization toolkit: Dynamic range quantization [Online; accessed 2024-09-19]. https://www.tensorflow.org/model_optimization/guide/quantization

TensorFlow. (2024). Tensorflow lite: An easy way to deploy machine learning models on mobile and edge devices. https://www.tensorflow.org/lite

TensorFlow Lite. (2023). Tensorflow lite | machine learning for mobile and edge devices.

Tensorflow lite converter [Accessed: 2024-09-27]. (2024).

Tesla. (2019). Tesla autonomy day.

Tesla. (2021). Artificial intelligence & autopilot.

Thompson, N., Greenewald, K., Lee, K., & Manso, G. F. (2020). The computational limits of deep learning. *arXiv preprint arXiv:2007.05558*. https://doi.org/10.48550/arXiv.2007.05558

Tirthapura, S. K. (2017). Nvdla primer.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., et al. (2023). Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*. https://doi.org/10.48550/arXiv.2302.13971

Tramer, F., et al. (2016). Stealing machine learning models via prediction apis. *25th USENIX Security Symposium*, 601–618.

Truong, N. B., Sun, K., Lee, G. M., & Guo, Y. (2019). Gdpr-compliant personal data management: A blockchain-based solution. *IEEE Transactions on Information Forensics and Security*, 15, 1746–1761.

Tung, L. (2018). Data privacy: Why home robots could be this generation's privacy nightmare.

Union, E. (2016a). General data protection regulation (gdpr). *Official Journal of the European Union*. https://eur-lex.europa.eu/eli/reg/2016/679/oj

Union, E. (2016b). General data protection regulation (gdpr) [Accessed: 2024-09-20].

Upton, E., & Halfacree, G. (2014). *Raspberry pi user guide*. Wiley.

U.S. Department of Health & Human Services. (2013a). Summary of the hipaa privacy rule.

U.S. Department of Health & Human Services. (2013b). Summary of the hipaa privacy rule.

U.S. Department of Justice. (2020). Paige thompson indictment.

Venkataramani, S., et al. (2020). Efficient ai inference with self-awareness and self-optimization on edge devices. *IEEE Design & Test*, 37(3), 15–23.

Verbraeken, J., Wolting, M., Katzy, J., Klous, S., & Sips, R.-J. (2020). A survey on distributed machine learning. *ACM Computing Surveys (CSUR)*, 53(2), 1–33.

Vohs, K. D., et al. (2008). Making choices impairs subsequent self-control: A limited-resource account of decision making, self-regulation, and active initiative. *Journal of Personality and Social Psychology*, 94(5), 883–898.

Voigt, P., & Von dem Bussche, A. (2017). *The eu general data protection regulation (gdpr): A practical guide*. Springer International Publishing.

Vorobeychik, Y., & Kantarcioglu, M. (2018). *Adversarial machine learning* (Vol. 12).

Wan, J., Tang, S., Li, D., Wang, S., Liu, C., & Abbas, H. (2018a). A manufacturing big data solution for active preventive maintenance. *IEEE Transactions on Industrial Informatics*, 13(4), 2039–2047.

Wan, J., Tang, S., Li, D., Wang, S., Liu, C., & Abbas, H. (2018b). A manufacturing big data solution for active preventive maintenance. *IEEE Transactions on Industrial Informatics*, 13(4), 2039–2047.

Wang, C., Liang, Z., Wu, F., Cao, X., & Wu, D. (2018). Edge caching at base stations with device-to-device offloading. *IEEE Access*, 6, 16649–16657. https://doi.org/10.1109/ACCESS.2018.2810864

Wang, J., & Wang, H. (2018). Protecting intellectual property of deep neural networks with watermarking. *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, 159–172.

Wang, M., et al. (2022). E2train: Energy-efficient training of dnns with (mostly) integer operations. *Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, 770–785.

Wang, N., et al. (2021). Energy-efficient edge ai: A survey of algorithms, hardware, and opportunities. *IEEE Internet of Things Journal*, 8(8), 6399–6422.

Wang, Q., & Duan, Y. (2020). Enhancing reliability in edge computing with caching and redundancy. *IEEE Transactions on Industrial Informatics*, 16(6), 4290–4298.

Wang, S., & Krishnan, E. (2018a). Mobile health technology: A new paradigm for healthcare. *IEEE Consumer Electronics Magazine*, 7(5), 53–57. https://doi.org/10.1109/MCE.2018.2851737

Wang, S., & Krishnan, E. (2018b). Mobile health technology: A new paradigm for healthcare. *IEEE Consumer Electronics Magazine*, 7(5), 53–57.

Wang, S., Zhang, X., Zhang, Y., Wang, L., Yang, J., & Wang, W. (2017). A survey on mobile edge networks: Convergence of computing, caching and communications. *IEEE Access*, 5, 6757–6779. https://doi.org/10.1109/ACCESS.2017.2685434

Wang, S., Zhou, Q., & Chen, X. (2020). An overview of liquidity pools in decentralized exchanges. *IEEE Access*, 8, 181749–181757.

Wang, S., & Krishnan, E. (2018c). Mobile health technology: A new paradigm for healthcare. *IEEE Consumer Electronics Magazine*, 7(5), 53–57.

Wang, Y., & Su, X. (2019). Ensuring data integrity in decentralized edge networks. *IEEE Transactions on Network Science and Engineering*, 6(4), 826–839. https://doi.org/10.1109/TNSE.2019.2909921

Wang, Y., et al. (2015). A survey of 5g network: Architecture and emerging technologies. *IEEE Access*, 3, 1206–1232.

Wang, Z., Liu, B., Gong, Z., Hu, S., Xie, Y., & Zhang, W. (2020). High-performance and energy-efficient neural network inference with processing-in-memory architecture. *IEEE Transactions on Computers*, 69(9), 1352–1365.

Warden, P. (2018). Speech commands: A dataset for limited-vocabulary speech recognition.

Warden, P. (2019). The machine learning and ai community's first look at tensorflow lite. *TensorFlow Blog*. https://blog.tensorflow.org/2019/03/tensorflow-lite-for-mobile-and-edge-devices.html

Warden, P., & Situnayake, D. (2019). Tinyml: Enabling on-device machine learning in ultra-low-power microcontrollers. *Proceedings of the NeurIPS Workshops*.

Waterman, A., & Asanović, K. (2017). *The risc-v instruction set manual, volume i: Unprivileged isa* (tech. rep.). EECS Department, UC Berkeley.

Wei, J., & Zou, K. (2019). Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *Proceedings of the ACL*.

Wen, T.-H., et al. (2015). Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *arXiv preprint arXiv:1508.01745*.

White, T. (2012). *Hadoop: The definitive guide*. O'Reilly Media.

Winter, J. (2008). Trusted computing building blocks for embedded linux-based arm trustzone platforms. *Proceedings of the 3rd ACM Workshop on Scalable Trusted Computing*, 21–30.

Wolf, S., et al. (2021). Compressing deep neural networks via layer fusion. *arXiv preprint arXiv:2102.06515*.

Wong, E., & Kolter, J. Z. (2018). Provable defenses against adversarial examples via the convex outer adversarial polytope. *International Conference on Machine Learning*, 5286–5295.

Wooldridge, M. (2009). *An introduction to multiagent systems*. John Wiley & Sons.

Woolley, A. W., & Malone, T. W. (2011). What makes a team smart? more brains or more connections? *Harvard Business Review*, 89(6), 92–98.

Wu, B., Dai, X., Zhang, P., Wang, Y., & Sun, F. (2021). Visual and linguistic knowledge transfer for large scale semi-supervised object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(7), 2493–2506.

Wu, B., Dai, X., Zhang, P., Wang, Y., Sun, F., Wu, Y., Vajda, P., Jia, Y., Keutzer, K., et al. (2019). Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wu, J., et al. (2016). Quantized convolutional neural networks for mobile devices. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4820–4828.

Wu, J., et al. (2020). Ai at the edge: Neural network acceleration and mobile ai applications. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 67(11), 2767–2771.

Wu, M., et al. (2020). Integer quantization for deep learning inference: Principles and empirical evaluation. *arXiv preprint arXiv:2004.09602*.

Wu, W., et al. (2020). Lite transformer with long-short range attention.

Wu, Z., et al. (2021). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1), 4–24.

Xiao, B., & Benbasat, I. (2007). E-commerce product recommendation agents: Use, characteristics, and impact. *MIS Quarterly*, 31(1), 137–209.

Xu, C., Liu, Z., & Chen, H. (2021). Bridging industry and academia in edge ai: A collaborative approach. *IEEE Access*, 9, 14598–14608.

Xu, C., Liu, Z., Li, W., Zhang, H., & Zhang, Y. (2021). Energy-efficient inference for deep learning services in 5g edge networks. *IEEE Transactions on Network and Service Management*, 18(2), 2167–2180.

Xu, J., et al. (2020). Privacy-preserving federated brain tumour segmentation. *Machine Learning for Health*, 1–12.

Xu, L. D., et al. (2014). Internet of things in industries: A survey. *IEEE Transactions on Industrial Informatics*, 10(4), 2233–2243.

Xu, W., Zhang, K., Zhou, Y., Chen, X., & Li, X. (2019a). Health monitoring and management using internet of things (iot) sensing with cloud-based processing: Opportunities and challenges. *IEEE Network*, 33(6), 27–33. https://doi.org/10.1109/MNET.001.1900085

Xu, W., Zhang, K., Zhou, Y., Chen, X., & Li, X. (2019b). Health monitoring and management using internet of things (iot) sensing with cloud-based processing: Opportunities and challenges. *IEEE Network*, 33(6), 27–33.

Xu, X., Li, J., & Zhang, W. (2020). Edge computing resource allocation based on decentralized finance models. *IEEE Access*, 8, 150324–150333.

Xu, X., Liu, C., Zhang, K., Li, Y., & Peng, K. (2018). A survey on edge computing for the internet of things. *Electronics*, 7(12), 113. https://doi.org/10.3390/electronics7080113

Yan, Z., et al. (2018). Data privacy protection mechanisms in iot-based intelligent healthcare systems. *IEEE Communications Magazine*, 56(4), 64–69.

Yang, L., et al. (2019). Security and privacy of edge ai in cyber-physical systems. *IEEE Network*, 33(5), 150–156.

Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology*, 10(2), 12.

Yang, Q., Liu, Y., Cheng, Y., Kang, Y., Chen, T., & Yu, H. (2019). *Federated learning* (Vol. 13). Synthesis Lectures on Artificial Intelligence; Machine Learning.

Yang, Z., & Yu, M. (2017). Face recognition attendance system based on real-time video processing. *2017 13th IEEE International Conference on Electronic Measurement & Instruments (ICEMI)*, 697–701. https://doi.org/10.1109/ICEMI.2017.8265850

Yi, S., et al. (2015). Fog computing: Platform and applications. *2015 Third IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb)*, 73–78.

Yin, D., Chen, Y., Kannan, R., & Bartlett, P. (2018). Byzantine-robust distributed learning: Towards optimal statistical rates. *Proceedings of the 35th International Conference on Machine Learning*, 5650–5659.

Yiu, J. (2013). *The definitive guide to arm cortex-m3 and cortex-m4 processors*. Newnes.

Zagoruyko, S., & Komodakis, N. (2016). Wide residual networks. *Proceedings of the British Machine Vision Conference (BMVC)*.

Zanella, A., et al. (2014). Internet of things for smart cities. *IEEE Internet of Things Journal*, 1(1), 22–32.

Zhan, S., & Kojima, F. (2017). Low latency and high reliability 5g communications in vr/ar applications. *2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 343–348. https://doi.org/10.1109/INFCOMW.2017.8116417

Zhan, Y., Liu, Y., Gong, Y., Yang, K., & Wu, Q. (2020). A learning-based incentive mechanism for federated learning. *IEEE Internet of Things Journal*, 7(7), 6360–6368.

Zhang, C., et al. (2019). Precision agriculture in the 21st century: Geospatial and information technologies in crop management. *Agricultural Engineering International: CIGR Journal*, 21(1), 1–10.

Zhang, C., & Zhu, L. (2021). Blockchain-based federated learning: Methods, applications, and open challenges. https://arxiv.org/abs/2101.07583

Zhang, F., et al. (2019). Detecting adversarial examples via modeling layer behaviors.

Zhang, J., & Chen, K. (2020a). Joint latency and reliability optimization for multi-access edge computing in 5g networks. *IEEE Transactions on Wireless Communications*, 19(7), 4715–4728. https://doi.org/10.1109/TWC.2020.2988330

Zhang, J., & Tao, D. (2020). Empowering things with intelligence: A survey of the progress, challenges, and opportunities in artificial intelligence of things. *IEEE Internet of Things Journal*, 8(10), 7789–7817. https://doi.org/10.1109/JIOT.2020.3010208

Zhang, J., & Chen, K. (2020b). Joint latency and reliability optimization for multi-access edge computing in 5g networks. *IEEE Transactions on Wireless Communications*, 19(7), 4715–4728. https://doi.org/10.1109/TWC.2020.2988330

Zhang, J., et al. (2019). Adaptive traffic signal control for large-scale urban road networks. *Transportation Research Part C: Emerging Technologies*, 109, 44–59.

Zhang, K., et al. (2016a). Energy-efficient offloading for mobile edge computing in 5g heterogeneous networks. *IEEE Access*, 4, 5896–5907.

Zhang, K., Mao, Y., Leng, S., He, Y., & Zhang, Y. (2016b). Mobile-edge computing for vehicular networks: A promising network paradigm with predictive offloading. *IEEE Vehicular Technology Magazine*, 12(2), 36–44. https://doi.org/10.1109/MVT.2016.2572460

Zhang, K., et al. (2020). Edge intelligence: Edge computing for internet of things. *IEEE Internet of Things Journal*, 7(8), 6948–6962.

Zhang, L., et al. (2016). Privacy-preserving data aggregation in mobile phone sensing. *IEEE Transactions on Information Forensics and Security*, 11(5), 980–992.

Zhang, L., & Wu, J. (2021). Dynamic pricing mechanisms in decentralized edge computing markets. *IEEE Transactions on Services Computing*, 14(5), 1376–1385.

Zhang, N., et al. (2018). Software defined space-air-ground integrated vehicular networks: Challenges and solutions. *IEEE Communications Magazine*, 55(7), 101–109.

Zhang, Q., Cheng, L., & Boutaba, R. (2010). Cloud computing: State-of-the-art and research challenges. *Journal of Internet Services and Applications*, 1(1), 7–18.

Zhang, R., et al. (2020). Privacy-preserving ai in finance: A federated learning approach. *IEEE Computational Intelligence Magazine*, 15(4), 80–88.

Zhang, X., et al. (2019). Neural processing units for mobile ai applications: A review. *IEEE Access*, 7, 181069–181098.

Zhang, Y., et al. (2020). Transformer transducer: A streamable speech recognition model with transformer encoders and rnn-t loss. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 7829–7833.

Zhao, H., et al. (2017). Lstm network: A deep learning approach for short-term traffic forecast. *IET Intelligent Transport Systems*, 11(2), 68–75.

Zhao, X., & Sun, Y. (2020). Resource staking in edge computing networks: A defi approach. *IEEE Transactions on Network Science and Engineering*, 7(4), 3241–3252.

Zhao, Z., Zhang, S., Chen, T., & Zhang, C. (2019). Improving neural network quantization without retraining using outlier channel splitting. *International Conference on Machine Learning (ICML)*.

Zheng, X., Martin, P., & Brohman, K. (2014). Cloud service negotiation: Constellation of integration and exchange contracts. *Proceedings of the 2014 IEEE International Conference on Services Computing*, 857–864.

Zhou, G., et al. (2019). Edge intelligence: Paving the last mile of artificial intelligence with edge computing. *Proceedings of the IEEE*, 107(8), 1738–1762.

Zhu, L., Hu, L., Lin, J., Wang, W.-C., Chen, W.-M., & Han, S. (2023). Pockengine: Sparse and efficient fine-tuning in a pocket. *Proceedings of the IEEE/ACM International Symposium on Microarchitecture (MICRO)*.

Zhu, M., & Gupta, S. (2018). To prune, or not to prune: Exploring the efficacy of pruning for model compression. *arXiv preprint arXiv:1710.01878*.

Zoph, B., & Le, Q. V. (2017). Neural architecture search with reinforcement learning. *International Conference on Learning Representations (ICLR)*.