A Review on Decentralized Artificial Intelligence in the Era of Large Models

AARON Y., Crynux AI Lab, USA

Decentralized Artificial Intelligence (DeAI) represents a paradigm shift in AI development, aiming to distribute control and resources across a broader network of stakeholders. From the vantage points of data providers, computing powers, and model trainers, we delve into the intricacies of participant involvement in DeAI, elucidating the associated risks, challenges, and corresponding solutions. Furthermore, we shed light on the emergent challenges stemming from large-scale models, particularly in the realms of cryptography, privacy preservation, and network scalability. With its comprehensive coverage, this survey serves as an indispensable resource for researchers and practitioners navigating the dynamic terrain of DeAI. A more detailed and continuously updated version is available at https://deai.gitbook.io

CCS Concepts: • **Computing methodologies → Distributed artificial intelligence**.

Additional Key Words and Phrases: Decentralized AI, Federated Learning, Cryptography, Blockchain

ACM Reference Format:
Aaron Y.. 2024. A Review on Decentralized Artificial Intelligence in the Era of Large Models. *J. ACM* 37, 4, Article 111 (April 2024), 24 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Over the past decade, the field of Artificial Intelligence (AI) has experienced unprecedented growth, marked by remarkable achievements like ChatGPT, a product leveraging generative pre-trained transformers, which has expanded the horizons of AI capabilities, edging us closer to the theoretical concept of Artificial General Intelligence (AGI) – a machine endowed with human-like cognitive capacities. Nevertheless, this surge in advancement has raised significant apprehensions regarding the concentration of AI development.

The ascent of AI has prompted concerns about its centralization. Currently, AI research predominantly orbits a handful of formidable corporations and governmental bodies. These entities wield extensive training data, expensive computational resources, and substantial financial backing to drive pioneering research. While this centralization undeniably propels progress, it also presents pivotal challenges[23]:

- Monopolization: The concentration of power among a few dominant entities, endowed with extensive data and computational resources, poses a threat to competition. This concentration can stifle innovation by limiting the diversity of perspectives and methodologies crucial for progress. Consequently, there's a risk of homogenizing AI development, potentially hindering breakthroughs in pivotal domains.
- Privacy and Security Risks: Centralized servers provoke significant anxieties regarding data privacy. The consolidation of power may incentivize the creation of AI models that prioritize the interests of those in control, potentially infringing upon individual liberties.
- Lack of transparency and accountability: While regulatory frameworks like GDPR[46] exist to safeguard data privacy, there's a notable absence of external mechanisms to verify how companies internally utilize data and

Author's address: Aaron Y., c@crynux.ai, Crynux AI Lab, Mountain View, CA, USA, 94043.

train models. This lack of transparency and accountability undermines trust in centralized AI systems and raises concerns about ethical data practices.

- Incentive Mechanisms: Enterprises that reap the benefits of AI often fail to share these rewards with the users who contribute data and provide evaluation feedback. Moreover, current large language models (LLMs) consume vast amounts of data [169], potentially including high-quality public data without adequate compensation for contributors. Introducing incentive mechanisms to encourage greater participation in data sharing and evaluation processes is essential for enhancing model performance and fostering a more equitable AI ecosystem.

In response to these concerns, there is a burgeoning movement towards decentralized AI (DeAI) development. This paradigm advocates for the dispersion of control and resources across a broader network of participants, fostering enhanced transparency, accountability, and inclusivity. DeAI endeavors to construct AI models within a decentralized infrastructure, where data providers and computing resources are distributed. Importantly, it aims to ensure the data privacy, and model security during training and inference processes.

While decentralized AI is still in its infancy, it holds immense potential for shaping the future of the field. By nurturing a more democratic and fair approach to AI development, we can ensure that this transformative technology serves the betterment of all humanity.

1.1 Motivation

With the advent of groundbreaking models like ChatGPT and Sora, the AI landscape has undergone significant evolution, ushering in a new era fraught with novel challenges that warrant careful consideration. Intriguingly, existing reviews and surveys often narrowly equate Decentralized AI (DeAI) with blockchain applications in AI, or delve into specific technical approaches, such as cryptography and privacy preservation, within the realm of deep learning. However, these reviews frequently overlook the nuanced techniques, and lack focus on solutions to challenges posed by deAI.

DeAI intersects deep learning, cryptography, and network technologies, yet researchers within each domain often lack insight into cutting edge progress from others. Hence, this review aims to bridge these knowledge gaps and disseminate the latest advancements in this arena. Acknowledging the rapid pace of innovation, we concede our inability to cover all recent developments and instead encourage readers to explore the continually updated resource at https://deai.gitbook.io. Contributions to this collaborative effort are warmly welcomed.

The contributions of this review can be distilled as follows:

- Introducing a systematic definition of DeAI, a novel contribution to the field.
- Identifying and discussing the emerging DeAI challenges posed by large-scale models.
- Conducting a comprehensive survey of these challenges, exploring various methodologies and their analyses and comparisons.
- Investigating the problem from multifaceted perspectives, encompassing deep learning, cryptography, network and economics domains.

1.2 Article Organization

In this survey of decentralized AI (DeAI), we structure our discussion as follows:

In Section 2, we present a comprehensive definition of DeAI and delineate the resolved and outstanding challenges within the field.

Section 3 delves into the privacy risks inherent in DeAI, particularly their implications for data providers. We survey various cryptographic and privacy-preserving machine learning approaches aimed at mitigating these risks.

In Section 4, we scrutinize different types of attacks targeting model training processes, which pose security threats to trainers. We comprehensively review defensive techniques against these attacks.

Section 5 focuses on exploring mechanisms to incentivize DeAI participants to contribute high-quality data and services, while thwarting cheating behavior.

Section 6 investigates techniques for verifying computations to prevent malicious actors from yielding fake results, thereby safeguarding the integrity of DeAI processes.

Lastly, in Section 7, we address the challenges posed to network bandwidth by large-scale models, offering insights into potential solutions.

1.3 General Terminology

Throughout this paper, we employ numerous acronyms for the sake of brevity and convention. Table 1 provides a comprehensive list of these acronyms, aiding in the clear representation of concepts, models, methods, and algorithms discussed.

Table 1. List of acronyms

Acronym	Explanation
AI	Artificial Intelligence
DeAI	Decentralized Artificial Intelligence
LLM	Large Language Model
PII	Personally Identifiable Information
FL	Federated Learning
DP	Differential Privacy
MPC	Multi-Party Computation
TEE	Trusted Execution Environment
FHE	Fully Homomorphic Encryption
ZKP	Zero-Knowledge Proof
MIA	Membership Inference Attack
PEFT	Parameter Efficient Fine Tuning

2 PROBLEM DEFINITION

Figure 1 illustrates a simplified yet characteristic workflow of the deep learning model paradigm:

(1) Data providers contribute data for training purposes.
(2) Model training scripts are executed on computing resources.
(3) Upon model convergence, the model is stored and hosted at a specified location.
(4) Users submit input requests to utilize the model.
(5) The model host conducts model inference on computing resources and delivers the results to the user.

In a centralized AI scenario, the entity conducting the model training task typically centralizes and stores all data internally, thereby assuming responsibility for data storage, infrastructure procurement (either through ownership or rental), as well as model training, hosting, and inference functionalities. This centralized model not only consolidates

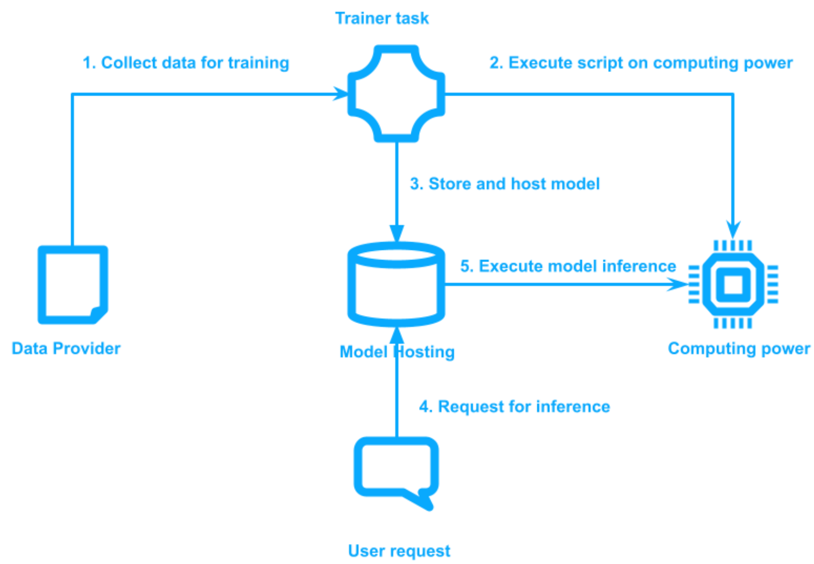

Fig. 1. Workflow of Deep Learning Paradigm

control but also raises concerns regarding data privacy and security, as well as potential biases inherent in the centralized decision-making process.

Contrastingly, in a fully decentralized AI setting, each of these participants—data providers, model trainers, and infrastructure owners—operates within distinct entities. This decentralized structure aims to distribute control and responsibility across a broader network of stakeholders, thereby mitigating the risks associated with centralization, such as monopolization and single points of failure. However, the lack of trust between parties in a decentralized environment poses its own set of challenges, including the need for robust mechanisms for data sharing, model training, and inference coordination while ensuring data privacy, security, and fairness.

This shift towards decentralization in AI research is motivated by the desire to foster transparency, accountability, and inclusivity in the development and deployment of AI systems. By distributing control and ownership of AI resources among diverse stakeholders, decentralized AI endeavors to democratize access to AI technologies and empower individuals and communities to actively participate in shaping the future of AI. Nonetheless, realizing the full potential of decentralized AI requires overcoming various technical, organizational, and regulatory challenges, as well as establishing trust and collaboration among stakeholders.

2.1 Data Providers

Current LLMs (e.g., GPT-4[3], Gemini[166], Llama[169]) are trained with data of trillions of tokens, indicating the imminent depletion of high-quality data [176]. To enhance model capability, incorporating more high-quality private data is imperative. From the perspective of data providers, concerns arise regarding privacy disclosure via data sharing, as data is utilized by the model training task and executed on computing powers. Importantly, models trained on their data should not disclose privacy, such as personally identifiable information (PII). Furthermore, incentivizing mechanisms are crucial to encourage data providers to produce more high-quality data. However, the current model training process lacks incentivization, failing to reward high-quality data via model monetization.

2.2 Model Training Task

The model training task trains data on computing power to obtain a model checkpoint, storing it on a model host platform. In a centralized solution, model training tasks can select relevant data for their use case, clean, and deduplicate data to enhance quality. In a decentralized framework, ensuring data relevance and quality without disclosing content poses challenges, particularly as data providers may be incentivized to produce more data. Moreover, data providers may conceal harmful information within data, potentially polluting the model. Additionally, as computing power is not owned by the model training task, decentralized AI frameworks may source computing power from crowd-sourcing, complicating the provision of stable and reliable computing services. Decentralized computing power necessitates mechanisms to prove computation on given data, ensuring system error tolerance and low latency.

2.3 Computing Power

Decentralized computing power entails computing nodes owned by different entities and organized via the internet. Large models consuming extensive data require increased network bandwidth, posing a significant challenge. Moreover, large models typically utilize multiple AI accelerators for computation. While Nvidia leverages PCIe to achieve 800GBps data transmission across chips, decentralized computing nodes connected via the internet face bandwidth limitations of 100GBps or lower, sometimes unreliable. Bandwidth emerges as a critical issue in the era of large models.

2.4 Challenges in Decentralized AI

Achieving a fully Decentralized AI (DeAI) framework entails overcoming a series of intricate challenges:

- **Privacy Preservation**: Preserving privacy while effectively utilizing data for model training poses a significant challenge. It necessitates sophisticated techniques to assess data quality and train models without divulging sensitive data, ensuring that model outputs do not contain privacy-compromising information.
- **Incentivization Mechanisms**: Encouraging data providers to contribute high-quality data requires the implementation of robust incentivization mechanisms. These mechanisms should adequately compensate data contributors while also rewarding model trainers and other participants for their valuable contributions.
- **Verification of Computation**: Establishing trust among participants in the DeAI network demands reliable mechanisms to verify computations. These mechanisms should ensure that all participants adhere to the agreed-upon protocols and accurately execute computations, thereby enhancing the overall integrity of the network.
- **Network Scalability**: Overcoming bandwidth limitations is crucial for enabling the complete implementation of decentralized computing power, particularly for large-scale models with billions of parameters and the network with thousands of remote nodes. Addressing scalability challenges involves optimizing network architectures, improving communication protocols, and leveraging advanced networking technologies.

2.5 Disambiguation

It's essential to clarify the scope of Decentralized AI (DeAI) within this survey. In this survey, we only focus on challenges and solutions from decentralized settings for deep learning model training and inference.

While the term "decentralized AI" is sometimes used to describe multi-agent systems [40, 126], this survey specifically focuses on the decentralized training and inference processes involving participants contributing data, computing power, and training scripts.

Furthermore, while many decentralized AI technologies utilize distributed computing techniques, such as data parallelism and model parallelism[175], distribution alone does not guarantee decentralization. These distribution techniques typically rely on implicit trust between nodes within a centralized setting. Decentralization, in addition to distribution, requires the establishment of trust mechanisms to prevent malicious attacks and incentive mechanisms to foster high-quality engagement and network improvement.

3 PRIVACY PRESERVATION

Privacy serves as a foundational principle in Decentralized AI (DeAI) systems. Data providers identify privacy leakage as their primary concern, highlighting the need for a trust mechanism to uphold their confidentiality in DeAI. The privacy preservation becomes more important in the context of handling powerful large models[144]. These models, while offering increased capacity and capability, can become vulnerable to privacy leaks during inference due to their very nature.

Large language models demonstrate remarkable memorization capabilities [28] and the capacity to learn from minimal data samples through techniques such as few-shot prompting[21]. While this adaptability enables them to deliver impressive outcomes across various tasks, it simultaneously raises concerns regarding potential privacy breaches. The very nature of these models, with their vast parameter spaces and ability to internalize vast amounts of data, makes them susceptible to inadvertently disclosing sensitive user information during inference.

Moreover, the decentralized nature of AI exacerbates privacy risks. Traditional centralized models involve data being stored and processed in a single location, allowing for relatively easier implementation of privacy safeguards. In contrast, decentralized AI distributes data and computation across multiple nodes, complicating the task of ensuring privacy protection throughout the system.

Privacy attacks in DeAI systems can be categorized as follows[37, 135] :

- Network data leakage: Unique to DeAI, this leakage occurs when data is transmitted between computing nodes, posing challenges absent in centralized AI training or federated learning where data is located on computing power[149].
- Membership inference attack(MIA)[50, 75, 160, 173, 190]: With this attack, adversaries predict if a specific example is part of the training data based on inference outputs,
- Training data extraction: The superior memorization capability of large models renders them vulnerable to privacy leakage[76], particularly when personally identifiable information (PII) is included in training data. For instance, the study[29] extractes training data containing PII such as names, phone numbers, email addresses, IRC conversations, code snippets, and 128-bit UUIDs from GPT-2 inference[147].
- Gradient leakage[54, 207]: Despite computation occurring on devices with data in federated learning, privacy can still be compromised as local training data can be reconstructed from gradients [118, 183]. TAG[41] demonstrates the capability to reconstruct 88.9% tokens of private training data from gradient.
- Attribute inference attack: This attack infers personal attributes from text given at inference time. GPT-4 achieved 84% accuracy to predict users' personal attributes from their public posts in reddit[161]

In DeAI model training, data privacy preservation can be addressed in three stages:

- Data Preprocess: Techniques such as data filtering, noise introduction, anonymization, and data aggregation mitigate privacy risks locally before data transmission [184].
- Computation framework: Privacy preservation can be ingrained within the design of the computation framework. Federated Learning[90], for example, aggregates user private data to train models while ensuring protection by locally computing gradients without revealing user data. Additionally, cryptographic computation methods like Multi-party computation (MPC) [61] , Homomorphic Encryption (HE) [56] , and Trusted Execution Environment (TEE) [152] play a pivotal role in safeguarding user data privacy without divulging them to the computing nodes.
- Model training: Techniques like adversarial regularization[134], differential privacy [57], dropout, model stacking, and random deletion of neural connections[153] enhance privacy during model training.

3.1 Data Process

Data processing stands as a natural and effective strategy to mitigate privacy leakage during the preparation of pretraining corpora and finetuning instruction datasets. This process involves masking or filtering personally identifiable information (PII) and other sensitive data. However, such measures can inadvertently reduce diversity and lead to information loss, potentially weakening the capabilities of Large Language Models (LLMs) [185].

Moreover, while data anonymization techniques like k-anonymity[164], l-diversity [121], and t-closeness[104] can be employed to eliminate privacy information from data, they may not fully protect against membership inference attacks, as signatures other than PII may still identify data providers.

Additionally, deduplication[97] , although a simple and effective method, can improve model quality while mitigating privacy risks[85] associated with training data extraction and membership inference attacks .

3.2 Privacy Preserved Training

Various techniques applied in model training can also improve privacy preservation.

3.2.1 Differential privacy. Differential privacy[45], defined as ensuring that adjacent data cannot be distinguished, offers privacy protection through an information-theoretic guarantee[30]. In practice, differential privacy is implemented by adding noise to data[32], gradients[1], output[68], or objective functions[30]. However, while differential privacy is crucial for preserving privacy, it may blur long-tail examples in data distributions, resulting in reduced accuracy[82, 171], particularly for underrepresented groups[10, 48]. Despite its potential negative impact on pretrained model performance[8], differential privacy in fine-tune tasks can maintain model utility[108, 197].

3.2.2 Privacy Regularization. Privacy regularization [127] introduces penalties for generating privacy-sensitive information. For instance, PPLM[188] introduces instruction tuning with Direct Preference Optimization (DPO)[148] to reward generations that distinguish between publicly shareable and privacy-sensitive information.

3.3 Federated Learning

Federated Learning[90, 125] is a distributed machine learning paradigm that aggregates model gradients from decentralized data sources. In this paradigm, data providers compute gradients locally with a global model and local data, which are then shared with a coordinator. The coordinator manages the training process by distributing tasks to data providers and iteratively updating the model.

While traditionally the coordinator in federated learning is a centralized server, there are emerging studies exploring decentralized paradigms of serverless federated learning[93, 110, 120, 151].

While federated learning stands as one of the most practical paradigms for preserving privacy in machine learning, it encounters several challenges[107, 207]:

- Expensive Communication: The iterative communication between devices and the server for model updates significantly increases communication costs, especially for large models.
- Systems Heterogeneity: Devices exhibit vast differences in storage capacity, computational power, communication bandwidth, and availability. With millions of devices in the network, this heterogeneity leads to high levels of unreliability.
- Statistical Heterogeneity: Data volume, quality, and distribution vary widely across different devices, presenting challenges in achieving consistent model performance.
- Efficiency: Compared to centralized model training, federated learning entails additional costs due to communication overhead and device unavailability, resulting in longer training times.
- Privacy Concern: Despite federated learning's emphasis on privacy preservation, there are instances of privacy issues arising from gradient leakage, which remain a concern

3.4 Cryptographic Computation

3.4.1 Homomorphic Encryption. Homomorphic encryption (HE) enables arithmetic computation directly on ciphertext[2]. Data providers encrypt the input using a private key, and the results of computation remain encrypted. Fully homomorphic encryption (FHE) allows arbitrary operations on ciphertext but is less efficient[56]. By leveraging homomorphic encryption, neural networks can compute on encrypted data to protect data privacy [145]. However, homomorphic encryption requires a polynomial representation, whereas neural networks utilize non-linear layers for activation. Some methods approximate non-linear layers with polynomials[38, 59]. Nonetheless, the capability of neural networks relies on non-polynomial activations [99], leading to reduced accuracy in encrypted models. Moreover, homomorphic encryption introduces extraordinary latency increases [22, 117]., and to date, there's limited work utilizing homomorphic encryption on large models or evaluating it on large datasets.

3.4.2 Multi-Party Computation. Multi-Party Computation(MPC) enables multiple parties to collaborate on computations without disclosing their data to each other[195]. Based on MPC protocols, multiple servers can jointly train a model by secret sharing [6, 89, 129, 130, 150, 178]. However, this approach incurs high computational and communication overhead, especially for large models, which require significantly more computation and communication resources. Moreover, MPC protocols necessitate simultaneous coordination of all parties, which may contradict the decentralized nature of environments where parties are unreliable.

3.4.3 Trusted Execution Environment. Trusted Execution Environment (TEE) creates an isolated environment ensuring code authentication, runtime state integrity, and data confidentiality. Intel SGX is one of the most studied TEE solutions[36, 124], providing a trusted hardware mechanism to create protected containers called enclaves. However, current TEE solutions have limitations for deep learning models, including significant overhead for memory-intensive tasks, limited memory capacity (e.g., 128MB default in Intel SGX), and support for limited CPU instructions without GPU leverage. Efforts have been made to offload computationally intensive layers of deep learning models to the GPU while maintaining integrity and confidentiality within an enclave[170]. Despite these efforts, complicated implementations have led to discovered attacks to TEE [137].

Table 2. Privacy Preservation Methods

Method	Model Performance	Efficiency	Network Requirement	Risk
DP	Lower	Similar	Low	
FL	Similar	Slightly slower	High	Gradient leakage
FHE	Much lower	Much slower	Low	MIA
TEE	Same	Much slower	Low	MIA
MPC	Same	Much slower	Very high	MIA

3.5 Challenges of Privacy Preservation Methods

Each privacy preservation approach has its drawbacks. Cryptographic methods guarantee a high level of privacy but suffer from significant efficiency drawbacks and may not defend against membership inference attacks. Differential privacy and adversarial regularization mitigate privacy attacks to some degree without introducing additional computation costs but may impact model performance. These challenges are often mutually incompatible[192]. Furthermore, the strengths of these techniques can also be double-edged; cryptographic approaches prevent data leakage in communication but also hinder data auditing, potentially facilitating backdoor attacks and data poisoning. To mitigate these challenges, multiple techniques are often used together, with federated learning serving as a backbone to integrate with other techniques such as differential privacy, MPC[53, 172], and cryptographic methods[67, 128, 191]. Some studies introduce trusted third parties to address efficiency challenges.

4 SECURITY

Malicious actors within DeAI environments pose significant privacy and security concerns. While malicious computing nodes and training tasks can leak the privacy of data providers, malicious data providers may compromise the security of DeAI models [60]. Attacks targeting models and federated learning can both impact DeAI systems by reducing model quality or inducing models to output desired contents.

DeAI involves permissionless participants in the network, making it vulnerable to network attacks[119]. Common attack means from the perspective of network participants include:

- Byzantine attack: Malicious agents upload arbitrary updates to degrade training performance [47, 94].
- Sybil attack[43]: Attackers create multiple dummy participant accounts to gain larger influence.

In DeAI, attackers often focus on the model during its training or inference phases, aiming for either targeted or untargeted poisoning. Targeted poisoning involves attackers manipulating the model to produce desired outputs of their choosing. On the other hand, untargeted poisoning seeks to disrupt the convergence of the global model, diminish its accuracy, or even cause it to diverge from its intended behavior. Data poisoning and model poisoning are common methods employed by attackers to achieve these objectives.

4.1 Data Poisoning

Data poisoning involves introducing malicious data to manipulate model outputs towards the attacker's intention. It's effective in both general machine learning [143] and federated learning[168]. In DeAI environments, where data is provided by individuals, it's particularly vulnerable to such attacks.

For classification models, a prevalent technique in data poisoning is label-flipping[139], wherein honest training examples are systematically switched from one class label to another. Consequently, the affected model erroneously predicts these examples to the corresponding altered labels.

To defend data poisoning, several defense strategies have been proposed. Reject on Negative Impact (RONI) [14] measures the impact of each training example on the error rate, and remove those with large nagatvie impact. Additionally, loss functions [81] can be leveraged to detect and mitigate the influence of malicious data.

Data sanitization techniques offer another avenue for early identification of malicious data. Methods such as BERT embedding [179], activations analysis[31] and provenance information analysis [13] can aid in this detection process.

4.2 Model Poisoning

Model poisoning[156] manipulates the global model by injecting malicious updates. It's a type of backdoor attack that maintains model performance on evaluated tasks but is controlled by attackers on backdoor tasks. Federated learning is vulnerable to model poisoning attacks[11], due to the decentralized nature of its participants, wherein any participant can update their model with injected malicious behavior

To address these risks, various defense methods have been proposed, primarily focusing on Byzantine robust aggregation techniques[158]:

- Distance based methods: These techniques distinguish outliers based on their distance from other agents [18, 26, 51].
- Performance based methods:These approaches evaluate updates and lower the weight of poorly performed updates [27, 105, 189].
- Statistics based methods:Utilizing the statistics of updates to identify outliers is another strategy employed to mitigate model poisoning attacks [142, 196].

These defense mechanisms aim to enhance the resilience of federated learning systems against model poisoning attacks by effectively identifying and mitigating the impact of malicious updates.

4.3 Sybil Attack

Sybil attacks[43], a type of network attack, involve creating numerous fake identities to gain influence over the network. In DeAI scenarios, sybil attacks can result in a larger proportion of malicious nodes during model training, increasing the likelihood of successful data or model poisoning.

One effective mitigation strategy involves identifying attacks through the diversity of updates[51]. This approach relies on the observation that targeted attacks often generate similar gradients with less diversity, enabling the detection of anomalous behavior indicative of Sybil attacks.

Moreover, other defense mechanisms against Sybil attacks in network scenarios [100] prove effective in the context of DeAI:

- Trusted certification[4]: Centralized authorities ensure that each entity possesses a certificate for network participation, thereby mitigating the proliferation of fake identities.
- Resource testing[9]: Nodes undergo testing to assess their computing capability and network bandwidth, facilitating the detection of anomalies suggestive of Sybil attacks.
- Economic costs and fees[123]: Introducing fees for participation discourages Sybil attacks by rendering them economically unviable, thereby enhancing the security of the network.

4.4 Impact of Large Models

The extraordinary capabilities of Large Language Models (LLMs) render them even more susceptible to the aforementioned attacks [37]. Owing to their enhanced capacity, identifying malicious data becomes more challenging since the model's performance on evaluated tasks remains largely unaffected, even after the introduction of malicious data. Consequently, backdoor attacks [24, 84, 102, 136, 157, 193, 205], particularly data poisoning attacks[5, 92, 180, 203], become significantly easier to execute, allowing malicious actors to manipulate LLMs into generating desired outputs with specific triggers.

The success of backdoor attacks implies that the model possesses spare learning capacity[65]. In addition to the defense techniques mentioned earlier, LLMs can utilize fine-tuning[154] to overwrite neurons tuned for backdoor inputs or prune these neurons[113] to effectively mitigate the impact of backdoor attacks.

4.5 Responsibility

In addition to the previously mentioned vulnerabilities, LLMs are susceptible to a range of other challenges [83]. These include phenomena such as hallucination[77], misinformation, and various active attack techniques such as adversarial attacks[91, 209], prompt injecting[64, 116], and jailbreak attacks[35]. Furthermore, other generative AI models have demonstrated the ability to fabricate human faces and voices convincingly, raising concerns regarding their potential misuse for fraudulent purposes.

Generative AI models are being increasingly utilized to fabricate synthetic videos and voices, fueling the proliferation of deceptive news, fraudulent schemes, scams, and other criminal pursuits. Unlike traditional centralized settings, where model training is overseen and regulated by a single entity, DeAI environments lack centralized oversight. This decentralized nature renders models susceptible to exploitation by malicious actors who may seek to manipulate or misuse them for illicit purposes. This underscores the critical importance of implementing robust security measures and oversight mechanisms to mitigate the risks associated with the misuse of generative AI in DeAI environments.

5 INCENTIVE MECHANISM

Incentive mechanisms play a pivotal role in DeAI systems, not only in rewarding participants for superior performance but also in rendering attacks economically unviable. Effective incentive mechanisms has been extensively discussed in various decentralized contexts such as decentralized markets[131], peer-to-peer networks[80], computation resource management[44] and crowdsensing platforms[63]. Furthermore, the decentralized nature of DeAI allows for the design of specific incentive mechanisms tailored to particular use cases.

The incentive mechanisms in DeAI are primarily aimed at addressing two major challenges:

- Motivate and maintain participants for their high performance.
- Evaluate participants' contributions accurately and fairly.

- Problem Formulation: It involves determining how to formulate the incentive problem, whether as a game theory model, auction model, or other relevant models.
- Contribution Evaluation: In a permissionless decentralized network, assessing the contribution of all nodes is crucial, especially considering the presence of potentially malicious nodes. Various strategies need to be employed to evaluate contributions accurately while mitigating the influence of malicious actors.

5.1 Problem Formulation

The foundation of designing a fair and effective incentive mechanism lies in formulating the incentive problem using appropriate theoretical frameworks such as game theory models, auction models, or other relevant models[174].

One prominent theoretical framework utilized in Federated Learning is the Stackelberg game[162]. In Federated Learning with Stackelberg game, the task requester assumes the role of the leader, while the clients act as followers[87, 202]. Here, the leader announces a strategy aimed at maximizing model performance, while clients base their actions on the leader's strategy, focusing on maximizing their resource utility to receive rewards.

Another approach, Federated Learning using auction theory, treats the task requester as an auctioneer and the clients as bidders [95, 199]. In this setup, the task requester initiates a task, and clients submit bids along with their computing costs and available resources. The task requester then determines the winner, assigns the task, and rewards the selected client. Some approaches[42, 198] uses auction theory to help aggregators calculate the optimal set of clients to maximize model performance within a limited budget.

In addition to game theory and auction theory, alternative approaches exist for formulating incentive mechanisms. For instance, incentive mechanism can be formulated as a social welfare maximization problem [165]. Furthermore, survey studies highlight additional theoretical frameworks such as contest theory and contract theory[186].

5.2 Contribution Evaluation

In the landscape of DeAI, federated learning stands as a pivotal approach, leveraging data contributions from various providers to enhance model performance. The efficacy of federated learning hinges on the quality of contributed data. It is a key component in incentive mechanisms how to evaluate contribution of DeAI participants.

However, the decentralized nature of federated learning introduces vulnerabilities, notably the risk of malicious actors attempting to exploit the system for undeserved rewards. These attackers may engage in various fraudulent activities, such as submitting fake, redundant, or low-quality data to inflate their rewards.

To mitigate such risks and ensure the integrity of the federated learning process, researchers have proposed diverse methods to evaluate the quality of contribution. Data Shapley[58] is an equitable data valuation metric that quantifies the the contribution of individual data points to a learning task. Metrics such as training loss reduction and accuracy enhancement serve as pivotal benchmarks in evaluating the efficacy of participants' contributions within incentive mechanisms[42, 69]. These mechanisms not only incentivize data providers to offer high-quality data but also safeguard against fraudulent behavior.

5.3 Copyright

Data providers within the DeAI paradigm may have concerns regarding the unauthorized utilization or potential plagiarism of their data by model trainers or other data contributors.

While ensuring privacy preservation necessitates defenses against membership inference and backdoor attacks, data providers are also interested in methods to detect if their data has been utilized in model training, particularly through the detection of specific triggers.

Data watermarking emerges as a prevalent technique applicable to various deep learning frameworks, including federated learning[167, 194] and LLMs[88, 140, 155]. Among these techniques, intentional backdoor insertion stands out as a practical approach for data copyright detection, involving the introduction of noise or specific text as triggers, subsequently verifying the output embeddings for data ownership validation.

6 VERIFICATION OF COMPUTATION

In the decentralized infrastructure of DeAI, computing resources often come from untrusted third parties. Consequently, the verification of computations becomes imperative to ensure the integrity of the process, safeguarding against instances where remote computing nodes might produce erroneous or even deliberately falsified results. Failure to verify computations not only risks rewarding malicious nodes undeservedly but also poses threats to the integrity of the entire model training process, including vulnerabilities to sybil attacks and poisoning attacks.

While some studies have delved into the realm of verifiable computing[55, 200], these investigations may not encompass the latest advancements following the emergence of blockchain technology and cryptographic techniques. Incorporating insights from these domains could yield novel solutions capable of addressing the evolving challenges within decentralized AI ecosystems, ensuring the robustness and trustworthiness of computations conducted by remote nodes.

6.1 Computation on Smart Contract

Ethereum[187] offers smart contract functionality that is theoretically proven to be Turing complete. This has sparked interest in leveraging smart contracts for AI computations[106]. However, the practicality of utilizing smart contracts for AI computations is limited by the substantial gas costs associated with such operations, rendering them impractical for handling large models.

6.2 Zero-Knowledge Proof

Zero-Knowledge Proof (ZKP)[62] is a cryptographic technique enabling a prover to convince a verifier of a statement's truth without revealing any additional information beyond the validity of the statement itself.

One prominent instantiation of ZKP is zk-SNARKs (Zero-Knowledge Succinct Non-interactive ARgument of Knowledge)[17], which has been applied in machine learning domains [52, 204]. This novel paradigm facilitates the verification of AI computations, particularly in deep learning model inference scenarios [49, 86, 98], enabling computation offloading to untrusted devices while ensuring the integrity of the process.

However, in ZKP solutions, the translation of functions into arithmetic circuits entails high costs for proof generation. These costs can be as much as 1000 times greater than native computations, rendering ZKP solutions impractical for handling large models.

6.3 Blockchain Audit

Before the emergence of blockchain technology, early explorations were undertaken to construct audit-based solutions[16]. These solutions relied on trusted clients to recompute sampled tasks performed by untrusted workers, employing incentive mechanisms to reward honest work and penalize cheating.

The advent of blockchain technology, notably underlying Bitcoin[132], brought about revolutionary features such as immutability and traceability in decentralized data storage.

In the realm of federated learning, blockchain has been harnessed to ensure data provenance and maintain auditable blocks [12, 33, 110, 146]. This integration ensures the verification of learning processes, enabling validators to scrutinize results. Additionally, the incentive mechanisms inherent in blockchain ecosystems incentivize nodes to contribute high-quality data and services to decentralized systems.

6.4 Consensus Protocol

Consensus protocols utilize smart contracts to orchestrate verification workflows while distributing AI computations across decentralized devices. Crynux H-net[72] establishes a permissionless and serverless DeAI network, and verifies computation results by cross-validating with other nodes executing the same task. This approach involves two key steps:

(1) Nodes upload commitments, derived from hashed signatures of results using local private seeds, onto the blockchain.
(2) Following the submission of commitments by all nodes, they can then submit their results to the blockchain for verification by smart contracts.

This consensus protocol effectively circumvents collusion among nodes, eschews reliance on trusted validators, and avoids additional computation complexity, rendering it well-suited for handling large models in DeAI settings.

In addition to the aforementioned methods, Trusted Execution Environment (TEE) presents another avenue for verifying computations by executing authorized code within isolated environments [170].

7 NETWORK COMMUNICATION

DeAI leverages the internet infrastructure for facilitating communication among diverse parties. This communication framework is fundamental for Federated Learning (FL) and Multi-party Computation (MPC) within the DeAI paradigm. Both FL and MPC heavily rely on communication protocols that facilitate multiple rounds of interaction among participating nodes.

7.1 Optimizing Computation Protocol

Efficient network communication is critical for the success of FL and MPC protocols within DeAI settings[107]. However, the communication overhead associated with transmitting checkpoints and model updates can significantly impact the overall efficiency. The optimized design of computation protocols minimizes the number of communication rounds required between nodes, enhances the efficiency and scalability of DeAI systems.

7.1.1 Local updating. Local updating mechanisms emerge as crucial strategies for minimizing communication overhead between nodes, thereby optimizing network utilization. Local SGD[163] enables independent execution of stochastic gradient descent on multiple worker nodes in parallel, and achieves the convergence rate comparable to traditional mini-batch methods[39].

7.1.2 Cryptography. Most secure sharing protocols necessitate multi-round peer-to-peer communication, posing challenges particularly when dealing with models containing billions of parameters, where completing such communication in a reasonable timeframe becomes impractical.

To address this issue, optimization on protocols are essential, notably focusing on garbled-circuites MPC protocols[15, 122]. A key optimization strategy involves the design of compilers tailored to generate fewer garbled-circuit gates, thereby reducing the size of data transmissions and alleviating the burden on network communication. However, by far, there is no practical method applied to large models.

7.1.3 Distribution topology. Many techniques derived from distributed computing are applied in DeAI settings, offering innovative solutions to various challenges.

Decentralized training exhibits potential advantages over centralized counterparts, particularly in scenarios characterized by network constraints such as low bandwidth or high latency[111].

Parallelism strategies are pivotal for accelerating model training across multiple GPUs, addressing the computational challenges inherent in large-scale models.

Data parallelism stands as the most prevalent approach, wherein datasets are partitioned into subsets and distributed among workers, each equipped with a model replica. Here, each worker processes a mini-batch within its assigned subset, computing weight updates independently. Communication is essential to synchronize gradients computed across devices. Parameter servers [103] and AllReduce communication protocols[138] facilitate this synchronization.

As large models nowadays exceed the capacity of a single GPU, model parallelism[159] emerges as a solution, distributing different parts of the model across multiple GPUs simultaneously.

Pipeline parallelism[78, 133] optimizes computation by breaking it into stages and forming a pipeline, with each stage executed on a distinct device. This method enhances throughput by enabling parallel processing of micro-batches.

These parallelism strategies relies on GPU interconnect techniques, such as NVLink and PCI-E [101]. These techniques provide a high-bandwidth communication between GPUs that can be as high as 1000Gbps on a centralized server. However, the decentralized environment is built on internet with bandwidth of 10-100 Mbps.

Petal [19, 20] allocates different layers of the model to decentralized GPUs on 10-100 Mbps internet for inference and infetune BLOOM-176B model[96].

Studies are also made to pretrain and fine-tune foundation model on 500Mbps network[182, 201]. They optimize scheduling based on a communication matrix incorporating bandwidth and latency information between decentralized nodes. Despite 100x slower communication, their distributed training setups across 64 GPUs in 8 regions globally incur only a 1.7-3.5x slowdown compared to centralized data centers.

In the federated learning context, hierarchical topology[114] is introduced to optimize communication efficiency of FL. This topology leverages edge servers to aggregate updates. This approach effectively reduces the overall communication burden.

7.2 Compression

Model compression[34, 70] reduces the size of models to reduce the size of model, thereby facilitating efficient communication of model weights or gradients[112].

Quantization emerges as a prominent approach, wherein weights are quantized to lower bit precision[79]. This not only reduces model size but also enhances computational efficiency. Notable examples include DoReFa-Net[206], which employs 1-bit weights with 2-bit gradients .

Sparsification techniques focus on pruning weights with negligible impact on model performance, thereby reducing model capacity without significant loss in accuracy[208]. In distributed training scenarios, only gradients surpassing a threshold are communicated, leading to 99% savings in gradient exchange[7].

Federated Dropout[25] trains and updates smaller subnets of the model, thereby reducing both local computation costs and communication payloads in federated learning.

Furthermore, factorization techniques offer a means to decompose weight matrices into low-rank representations, thereby reducing the bandwidth required for communication[181].

These diverse strategies collectively contribute to optimizing model communication in decentralized settings.

7.3 Parameter Efficient Fine Tuning

Within the landscape of large-scale models, conventional compression techniques may prove insufficient to address the challenges posed by their large size. In this context, Parameter Efficient Fine Tuning (PEFT) emerges as a more aggressive strategy aimed at reducing the number of trainable weights, thereby mitigating communication overhead.

PEFT adjusts only a small proportion of model parameters, resulting in a significant reduction in computational complexity. This reduction in the number of trainable weights translates to a decrease in communication payload, particularly beneficial in Decentralized AI (DeAI) settings.

PEFT can be categorized into several approaches[71]:

- Additive modules modifies the model architecture by injecting an additive trainable modules or layers [73, 141].
- Soft prompts utilize continuous embedding spaces of soft prompts to refine model performance. Notable examples include Prefix-Tuning, which leverages prefix vectors for inference after fine-tuning [109, 115].
- Selective fine-tuning involves selecting a small subset of parameters, making them tunable while keeping the remaining weights frozen. Diff Pruning[66, 177] applies parameter pruning techniques to achieve efficiency gains .
- Reparameterized PEFT employs low-rank parameterization techniques to construct more efficient representations, which are then transformed back for inference[74].

The emergence of large-scale models has catalyzed the development of diverse techniques aimed at significantly reducing the computational complexity and communication overhead associated with these models in the realm of DeAI.

8 CONCLUSION

In this comprehensive review, we establish a systematic definition of Decentralized AI (DeAI) and meticulously examine the challenges and complexities inherent in achieving complete decentralization. We pioneer an exploration into the unique challenges posed by the advent of large-scale models, shedding light on their implications for DeAI ecosystems.

Our analysis delves into various critical domains:

- Data Privacy Preservation: We scrutinize the risks and challenges confronting data providers, presenting an list of techniques spanning privacy learning, federated learning, and cryptography to mitigate privacy concerns effectively.
- Security Attacks: Through a detailed examination, we dissect the list of potential attacks targeting model training in DeAI and survey existing solutions to fortify defenses against such threats.
- Incentive Mechanisms: We delve into strategies aimed at incentivizing data providers and computing powers to sustain high-quality service within the DeAI network, emphasizing the importance of fair evaluation mechanisms. In addition, we discussed the copyright protection techniques for data providers.
- Verification of Computation: Our analysis encompasses techniques designed to verify computation results from computing powers, crucial for safeguarding against fraudulent activities such as fake result attacks.
- Network Communication: We explore diverse solutions geared towards enhancing the efficiency of network communication within decentralized settings, including computation protocol, topology, compression, and parameter-efficient fine-tuning.

Moreover, we confront the dual nature of features exhibited by large models in DeAI:

- While large models exhibit extraordinary memorization capabilities, they also raise significant privacy concerns, particularly regarding the inadvertent memorization of sensitive information.
- Privacy preservation techniques serve as vital safeguards against privacy breaches, yet they may inadvertently obscure the source of origin and hinder copyright protection efforts.
- Data encryption techniques prevents data leakage during network communication, but they also present challenges in auditing malicious data from other parties.

Many existing solutions in the DeAI landscape may not be entirely feasible in the context of large models, given their larger size and enhanced memorization and generalization capabilities. Furthermore, due to the rapid evolution of this field and the vast scope for exploration, some important works may have been overlooked in this survey. For instance, the topic of model encryption warrants further investigation.

To stay updated with the latest version of this survey and to explore emerging topics further, we invite readers to visit our continuously updated repository at https://deai.gitbook.com. We also encourage researchers to collaborate on this open-source GitBook, contributing insightful information to enrich the content.

REFERENCES

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. 308–318.

[2] Abbas Acar, Hidayet Aksu, A Selcuk Uluagac, and Mauro Conti. 2018. A survey on homomorphic encryption schemes: Theory and implementation. *ACM Computing Surveys (Csur)* 51, 4 (2018), 1–35.

[3] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[4] Atul Adya, William J Bolosky, Miguel Castro, Gerald Cermak, Ronnie Chaiken, John R Douceur, Jon Howell, Jacob R Lorch, Marvin Theimer, and Roger P Wattenhofer. 2002. FARSITE: Federated, available, and reliable storage for an incompletely trusted environment. *ACM SIGOPS Operating Systems Review* 36, SI (2002), 1–14.

[5] Hojjat Aghakhani, Wei Dai, Andre Manoel, Xavier Fernandes, Anant Kharkar, Christopher Kruegel, Giovanni Vigna, David Evans, Ben Zorn, and Robert Sim. 2023. Trojanpuzzle: Covertly poisoning code-suggestion models. *arXiv preprint arXiv:2301.02344* (2023).

[6] Nitin Agrawal, Ali Shahin Shamsabadi, Matt J Kusner, and Adrià Gascón. 2019. QUOTIENT: two-party secure neural network training and prediction. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 1231–1247.

[7] Alham Fikri Aji and Kenneth Heafield. 2017. Sparse communication for distributed gradient descent. *arXiv preprint arXiv:1704.05021* (2017).

[8] Rohan Anil, Badih Ghazi, Vineet Gupta, Ravi Kumar, and Pasin Manurangsi. 2021. Large-scale differentially private BERT. *arXiv preprint arXiv:2108.01624* (2021).

[9] James Aspnes, Collin Jackson, and Arvind Krishnamurthy. 2005. *Exposing computationally-challenged Byzantine impostors*. Technical Report. Technical Report YALEU/DCS/TR-1332, Yale University Department of Computer ….

[10] Eugene Bagdasaryan, Omid Poursaeed, and Vitaly Shmatikov. 2019. Differential privacy has disparate impact on model accuracy. *Advances in neural information processing systems* 32 (2019).

[11] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. 2020. How to backdoor federated learning. In *International conference on artificial intelligence and statistics*. PMLR, 2938–2948.

[12] Xianglin Bao, Cheng Su, Yan Xiong, Wenchao Huang, and Yifei Hu. 2019. Flchain: A blockchain for auditable federated learning with trust and incentive. In *2019 5th International Conference on Big Data Computing and Communications (BIGCOM)*. IEEE, 151–159.

[13] Nathalie Baracaldo, Bryant Chen, Heiko Ludwig, and Jaehoon Amir Safavi. 2017. Mitigating poisoning attacks on machine learning models: A data provenance based approach. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*. 103–110.

[14] Marco Barreno, Blaine Nelson, Anthony D Joseph, and J Doug Tygar. 2010. The security of machine learning. *Machine Learning* 81 (2010), 121–148.

[15] Zuzana Beerliová-Trubíniová and Martin Hirt. 2008. Perfectly-secure MPC with linear communication complexity. In *Theory of Cryptography Conference*. Springer, 213–230.

[16] Mira Belenkiy, Melissa Chase, C Chris Erway, John Jannotti, Alptekin Küpçü, and Anna Lysyanskaya. 2008. Incentivizing outsourced computation. In *Proceedings of the 3rd international workshop on Economics of networked systems*. 85–90.

[17] Eli Ben-Sasson, Alessandro Chiesa, Eran Tromer, and Madars Virza. 2014. Succinct {Non-Interactive} zero knowledge for a von neumann architecture. In *23rd USENIX Security Symposium (USENIX Security 14)*. 781–796.

[18] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. 2017. Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in neural information processing systems* 30 (2017).

[19] Alexander Borzunov, Dmitry Baranchuk, Tim Dettmers, Max Ryabinin, Younes Belkada, Artem Chumachenko, Pavel Samygin, and Colin Raffel. 2022. Petals: Collaborative Inference and Fine-tuning of Large Models. *arXiv preprint arXiv:2209.01188* (2022). https://arxiv.org/abs/2209.01188

[20] Alexander Borzunov, Max Ryabinin, Artem Chumachenko, Dmitry Baranchuk, Tim Dettmers, Younes Belkada, Pavel Samygin, and Colin A Raffel. 2024. Distributed Inference and Fine-tuning of Large Language Models Over The Internet. *Advances in Neural Information Processing Systems* 36 (2024).

[21] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[22] Alon Brutzkus, Ran Gilad-Bachrach, and Oren Elisha. 2019. Low latency privacy preserving inference. In *International Conference on Machine Learning*. PMLR, 812–821.

[23] Erik Brynjolfsson and Andrew Ng. 2023. Big AI can centralize decision-making and power, and that'sa problem. *Missing links in ai governance* 65 (2023).

[24] Xiangrui Cai, Haidong Xu, Sihan Xu, Ying Zhang, et al. 2022. Badprompt: Backdoor attacks on continuous prompts. *Advances in Neural Information Processing Systems* 35 (2022), 37068–37080.

[25] Sebastian Caldas, Jakub Konečny, H Brendan McMahan, and Ameet Talwalkar. 2018. Expanding the reach of federated learning by reducing client resource requirements. *arXiv preprint arXiv:1812.07210* (2018).

[26] Di Cao, Shan Chang, Zhijian Lin, Guohua Liu, and Donghong Sun. 2019. Understanding distributed poisoning attack in federated learning. In *2019 IEEE 25th international conference on parallel and distributed systems (ICPADS)*. IEEE, 233–239.

[27] Xinyang Cao and Lifeng Lai. 2019. Distributed gradient descent algorithm robust to an arbitrary number of byzantine attackers. *IEEE Transactions on Signal Processing* 67, 22 (2019), 5850–5864.

[28] Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2022. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646* (2022).

[29] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*. 2633–2650.

[30] Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. 2011. Differentially private empirical risk minimization. *Journal of Machine Learning Research* 12, 3 (2011).

[31] Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy, and Biplav Srivastava. 2018. Detecting backdoor attacks on deep neural networks by activation clustering. *arXiv preprint arXiv:1811.03728* (2018).

[32] Rui Chen, Noman Mohammed, Benjamin CM Fung, Bipin C Desai, and Li Xiong. 2011. Publishing set-valued data via differential privacy. *Proceedings of the VLDB Endowment* 4, 11 (2011), 1087–1098.

[33] Xuhui Chen, Jinlong Ji, Changqing Luo, Weixian Liao, and Pan Li. 2018. When machine learning meets blockchain: A decentralized, privacy-preserving and secure design. In *2018 IEEE international conference on big data (big data)*. IEEE, 1178–1187.

[34] Tejalal Choudhary, Vipul Mishra, Anurag Goswami, and Jagannathan Sarangapani. 2020. A comprehensive survey on model compression and acceleration. *Artificial Intelligence Review* 53 (2020), 5113–5155.

[35] Junjie Chu, Yugeng Liu, Ziqing Yang, Xinyue Shen, Michael Backes, and Yang Zhang. 2024. Comprehensive assessment of jailbreak attacks against llms. *arXiv preprint arXiv:2402.05668* (2024).

[36] Victor Costan and Srinivas Devadas. 2016. Intel SGX explained. *Cryptology ePrint Archive* (2016).

[37] Badhan Chandra Das, M Hadi Amini, and Yanzhao Wu. 2024. Security and privacy challenges of large language models: A survey. *arXiv preprint arXiv:2402.00888* (2024).

[38] Roshan Dathathri, Blagovesta Kostova, Olli Saarikivi, Wei Dai, Kim Laine, and Madan Musuvathi. 2020. EVA: An encrypted vector arithmetic language and compiler for efficient homomorphic computation. In *Proceedings of the 41st ACM SIGPLAN conference on programming language design and implementation*. 546–561.

[39] Ofer Dekel, Ran Gilad-Bachrach, Ohad Shamir, and Lin Xiao. 2012. Optimal Distributed Online Prediction Using Mini-Batches. *Journal of Machine Learning Research* 13, 1 (2012).

[40] Yves Demazeau and J-P Müller. 1990. *Decentralized Ai*. Vol. 2. Elsevier.

[41] Jieren Deng, Yijue Wang, Ji Li, Chao Shang, Hang Liu, Sanguthevar Rajasekaran, and Caiwen Ding. 2021. Tag: Gradient attack on transformer-based language models. *arXiv preprint arXiv:2103.06819* (2021).

[42] Yongheng Deng, Feng Lyu, Ju Ren, Yi-Chao Chen, Peng Yang, Yuezhi Zhou, and Yaoxue Zhang. 2021. Fair: Quality-aware federated learning with precise user incentive and model aggregation. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 1–10.

[43] John R Douceur. 2002. The sybil attack. In *International workshop on peer-to-peer systems*. Springer, 251–260.

[44] K Eric Drexler and Mark S Miller. 1988. Incentive engineering for computational resource management. *The ecology of Computation* 2 (1988), 231–266.

[45] Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science* 9, 3–4 (2014), 211–407.

[46] European Parliament and Council of the European Union. [n. d.]. *Regulation (EU) 2016/679 of the European Parliament and of the Council*. https://data.europa.eu/eli/reg/2016/679/oj

[47] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Gong. 2020. Local model poisoning attacks to {Byzantine-Robust} federated learning. In *29th USENIX security symposium (USENIX Security 20)*. 1605–1622.

[48] Tom Farrand, Fatemehsadat Mireshghallah, Sahib Singh, and Andrew Trask. 2020. Neither private nor fair: Impact of data imbalance on utility and fairness in differential privacy. In *Proceedings of the 2020 workshop on privacy-preserving machine learning in practice*. 15–19.

[49] Boyuan Feng, Lianke Qin, Zhenfei Zhang, Yufei Ding, and Shumo Chu. 2021. Zen: An optimizing compiler for verifiable, zero-knowledge neural network inferences. *Cryptology ePrint Archive* (2021).

[50] Wenjie Fu, Huandong Wang, Chen Gao, Guanghua Liu, Yong Li, and Tao Jiang. 2023. Practical membership inference attacks against fine-tuned large language models via self-prompt calibration. *arXiv preprint arXiv:2311.06062* (2023).

[51] Clement Fung, Chris JM Yoon, and Ivan Beschastnikh. 2018. Mitigating sybils in federated learning poisoning. *arXiv preprint arXiv:1808.04866* (2018).

[52] Bianca-Mihaela Ganescu and Jonathan Passerat-Palmbach. 2024. Trust the Process: Zero-Knowledge Machine Learning to Enhance Trust in Generative AI Interactions. *arXiv preprint arXiv:2402.06414* (2024).

[53] Till Gehlhar, Felix Marx, Thomas Schneider, Ajith Suresh, Tobias Wehrle, and Hossein Yalame. 2023. Safefl: Mpc-friendly framework for private and robust federated learning. In *2023 IEEE Security and Privacy Workshops (SPW)*. IEEE, 69–76.

[54] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. 2020. Inverting gradients-how easy is it to break privacy in federated learning? *Advances in neural information processing systems* 33 (2020), 16937–16947.

[55] Rosario Gennaro, Craig Gentry, and Bryan Parno. 2010. Non-interactive verifiable computing: Outsourcing computation to untrusted workers. In *Advances in Cryptology–CRYPTO 2010: 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings 30*. Springer, 465–482.

[56] Craig Gentry. 2009. *A fully homomorphic encryption scheme*. Stanford university.

[57] Robin C Geyer, Tassilo Klein, and Moin Nabi. 2017. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557* (2017).

[58] Amirata Ghorbani and James Zou. 2019. Data shapley: Equitable valuation of data for machine learning. In *International conference on machine learning*. PMLR, 2242–2251.

[59] Ran Gilad-Bachrach, Nathan Dowlin, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing. 2016. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In *International conference on machine learning*. PMLR, 201–210.

[60] Micah Goldblum, Dimitris Tsipras, Chulin Xie, Xinyun Chen, Avi Schwarzschild, Dawn Song, Aleksander Mądry, Bo Li, and Tom Goldstein. 2022. Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 2 (2022), 1563–1580.

[61] Oded Goldreich. 1998. Secure multi-party computation. *Manuscript. Preliminary version* 78, 110 (1998), 1–108.

[62] Oded Goldreich and Yair Oren. 1994. Definitions and properties of zero-knowledge proof systems. *Journal of Cryptology* 7, 1 (1994), 1–32.

[63] Xiaowen Gong and Ness Shroff. 2018. Incentivizing truthful data quality for quality-aware mobile data crowdsourcing. In *Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*. 161–170.

[64] Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. 2023. Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*. 79–90.

[65] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. 2017. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733* (2017).

[66] Demi Guo, Alexander M Rush, and Yoon Kim. 2020. Parameter-efficient transfer learning with diff pruning. *arXiv preprint arXiv:2012.07463* (2020).

[67] Xiaojie Guo, Zheli Liu, Jin Li, Jiqiang Gao, Boyu Hou, Changyu Dong, and Thar Baker. 2020. V eri fl: Communication-efficient and fast verifiable aggregation for federated learning. *IEEE Transactions on Information Forensics and Security* 16 (2020), 1736–1751.

[68] Jihun Hamm, Yingjun Cao, and Mikhail Belkin. 2016. Learning privately from multiparty data. In *International Conference on Machine Learning*. PMLR, 555–563.

[69] Jingoo Han, Ahmad Faraz Khan, Syed Zawad, Ali Anwar, Nathalie Baracaldo Angel, Yi Zhou, Feng Yan, and Ali R Butt. 2022. Tokenized incentive for federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[70] Song Han, Huizi Mao, and William J Dally. 2015. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149* (2015).

[71] Zeyu Han, Chao Gao, Jinyang Liu, Sai Qian Zhang, et al. 2024. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608* (2024).

[72] Max Weng Young Wong Aaron Yuasa Henry Lee, Luke Weber. 2023. Crynux Hydrogen Network (H-Net): Decentralized AI Serving Network on Blockchain. *ResearchGate preprint doi:10.13140/RG.2.2.32697.54884* (2023).

[73] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *International conference on machine learning*. PMLR, 2790–2799.

[74] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685* (2021).

[75] Li Hu, Anli Yan, Hongyang Yan, Jin Li, Teng Huang, Yingying Zhang, Changyu Dong, and Chunsheng Yang. 2023. Defenses to membership inference attacks: A survey. *Comput. Surveys* 56, 4 (2023), 1–34.

[76] Jie Huang, Hanyin Shao, and Kevin Chen-Chuan Chang. 2022. Are large pre-trained language models leaking your personal information? *arXiv preprint arXiv:2205.12628* (2022).

[77] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232* (2023).

[78] Yanping Huang, Youlong Cheng, Ankur Bapna, Orhan Firat, Dehao Chen, Mia Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V Le, Yonghui Wu, et al. 2019. Gpipe: Efficient training of giant neural networks using pipeline parallelism. *Advances in neural information processing systems* 32 (2019).

[79] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. 2016. Binarized neural networks. *Advances in neural information processing systems* 29 (2016).

[80] Cornelius Ihle, Dennis Trautwein, Moritz Schubotz, Norman Meuschke, and Bela Gipp. 2023. Incentive mechanisms in peer-to-peer networks—a systematic literature review. *Comput. Surveys* 55, 14s (2023), 1–69.

[81] Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. 2018. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In *2018 IEEE symposium on security and privacy (SP)*. IEEE, 19–35.

[82] Bargav Jayaraman and David Evans. 2019. Evaluating differentially private machine learning in practice. In *28th USENIX Security Symposium (USENIX Security 19)*. 1895–1912.

[83] Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. 2023. Challenges and applications of large language models. *arXiv preprint arXiv:2307.10169* (2023).

[84] Nikhil Kandpal, Matthew Jagielski, Florian Tramèr, and Nicholas Carlini. 2023. Backdoor attacks for in-context learning with language models. *arXiv preprint arXiv:2307.14692* (2023).

[85] Nikhil Kandpal, Eric Wallace, and Colin Raffel. 2022. Deduplicating training data mitigates privacy risks in language models. In *International Conference on Machine Learning*. PMLR, 10697–10707.

[86] Daniel Kang, Tatsunori Hashimoto, Ion Stoica, and Yi Sun. 2022. Scaling up trustless dnn inference with zero-knowledge proofs. *arXiv preprint arXiv:2210.08674* (2022).

[87] Latif U Khan, Shashi Raj Pandey, Nguyen H Tran, Walid Saad, Zhu Han, Minh NH Nguyen, and Choong Seon Hong. 2020. Federated learning for edge networks: Resource optimization and incentive mechanism. *IEEE Communications Magazine* 58, 10 (2020), 88–93.

[88] John Kirchenbauer, Jonas Geiping, Yuxin Wen, Manli Shu, Khalid Saifullah, Kezhi Kong, Kasun Fernando, Aniruddha Saha, Micah Goldblum, and Tom Goldstein. 2023. On the reliability of watermarks for large language models. *arXiv preprint arXiv:2306.04634* (2023).

[89] Brian Knott, Shobha Venkataraman, Awni Hannun, Shubho Sengupta, Mark Ibrahim, and Laurens van der Maaten. 2021. Crypten: Secure multi-party computation meets machine learning. *Advances in Neural Information Processing Systems* 34 (2021), 4961–4973.

[90] Jakub Konecnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. 2016. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492* 8 (2016).

[91] Aounon Kumar, Chirag Agarwal, Suraj Srinivas, Soheil Feizi, and Hima Lakkaraju. 2023. Certifying llm safety against adversarial prompting. *arXiv preprint arXiv:2309.02705* (2023).

[92] Keita Kurita, Paul Michel, and Graham Neubig. 2020. Weight poisoning attacks on pre-trained models. *arXiv preprint arXiv:2004.06660* (2020).

[93] Anusha Lalitha, Shubhanshu Shekhar, Tara Javidi, and Farinaz Koushanfar. 2018. Fully decentralized federated learning. In *Third workshop on bayesian deep learning (NeurIPS)*, Vol. 2.

[94] Leslie Lamport, Robert Shostak, and Marshall Pease. 2019. The Byzantine generals problem. In *Concurrency: the works of leslie lamport*. 203–226.

[95] Tra Huong Thi Le, Nguyen H Tran, Yan Kyaw Tun, Zhu Han, and Choong Seon Hong. 2020. Auction based incentive design for efficient federated learning in cellular wireless networks. In *2020 IEEE wireless communications and networking conference (WCNC)*. IEEE, 1–6.

[96] Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2023. Bloom: A 176b-parameter open-access multilingual language model. (2023).

[97] Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2021. Deduplicating training data makes language models better. *arXiv preprint arXiv:2107.06499* (2021).

[98] Seunghwa Lee, Hankyung Ko, Jihye Kim, and Hyunok Oh. 2024. vcnn: Verifiable convolutional neural network based on zk-snarks. *IEEE Transactions on Dependable and Secure Computing* (2024).

[99] Moshe Leshno, Vladimir Ya Lin, Allan Pinkus, and Shimon Schocken. 1993. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural networks* 6, 6 (1993), 861–867.

[100] Brian Neil Levine, Clay Shields, and N Boris Margolin. 2006. A survey of solutions to the sybil attack. *University of Massachusetts Amherst, Amherst, MA* 7 (2006), 224.

[101] Ang Li, Shuaiwen Leon Song, Jieyang Chen, Jiajia Li, Xu Liu, Nathan R Tallent, and Kevin J Barker. 2019. Evaluating modern gpu interconnect: Pcie, nvlink, nv-sli, nvswitch and gpudirect. *IEEE Transactions on Parallel and Distributed Systems* 31, 1 (2019), 94–110.

[102] Linyang Li, Demin Song, Xiaonan Li, Jiehang Zeng, Ruotian Ma, and Xipeng Qiu. 2021. Backdoor attacks on pre-trained models by layerwise weight poisoning. *arXiv preprint arXiv:2108.13888* (2021).

[103] Mu Li, David G Andersen, Jun Woo Park, Alexander J Smola, Amr Ahmed, Vanja Josifovski, James Long, Eugene J Shekita, and Bor-Yiing Su. 2014. Scaling distributed machine learning with the parameter server. In *11th USENIX Symposium on operating systems design and implementation (OSDI 14)*. 583–598.

[104] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. 2006. t-closeness: Privacy beyond k-anonymity and l-diversity. In *2007 IEEE 23rd international conference on data engineering*. IEEE, 106–115.

[105] Suyi Li, Yong Cheng, Yang Liu, Wei Wang, and Tianjian Chen. 2019. Abnormal client behavior detection in federated learning. *arXiv preprint arXiv:1910.09933* (2019).

[106] Tao Li, Yaozheng Fang, Ye Lu, Jinni Yang, Zhaolong Jian, Zhiguo Wan, and Yusen Li. 2022. Smartvm: A smart contract virtual machine for fast on-chain dnn computations. *IEEE Transactions on Parallel and Distributed Systems* 33, 12 (2022), 4100–4116.

[107] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. 2020. Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine* 37, 3 (2020), 50–60.

[108] Xuechen Li, Florian Tramer, Percy Liang, and Tatsunori Hashimoto. 2021. Large language models can be strong differentially private learners. *arXiv preprint arXiv:2110.05679* (2021).

[109] Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190* (2021).

[110] Yuzheng Li, Chuan Chen, Nan Liu, Huawei Huang, Zibin Zheng, and Qiang Yan. 2020. A blockchain-based decentralized federated learning framework with committee consensus. *IEEE Network* 35, 1 (2020), 234–241.

[111] Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. 2017. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. *Advances in neural information processing systems* 30 (2017).

[112] Yujun Lin, Song Han, Huizi Mao, Yu Wang, and William J Dally. 2017. Deep gradient compression: Reducing the communication bandwidth for distributed training. *arXiv preprint arXiv:1712.01887* (2017).

[113] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. 2018. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *International symposium on research in attacks, intrusions, and defenses*. Springer, 273–294.

[114] Lumin Liu, Jun Zhang, SH Song, and Khaled B Letaief. 2019. Edge-assisted hierarchical federated learning with non-iid data. *arXiv preprint arXiv:1905.06641* (2019).

[115] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602* (2021).

[116] Yi Liu, Gelei Deng, Yuekang Li, Kailong Wang, Tianwei Zhang, Yepang Liu, Haoyu Wang, Yan Zheng, and Yang Liu. 2023. Prompt Injection attack against LLM-integrated Applications. *arXiv preprint arXiv:2306.05499* (2023).

[117] Qian Lou and Lei Jiang. 2021. HEMET: a homomorphic-encryption-friendly privacy-preserving mobile neural network architecture. In *International conference on machine learning*. PMLR, 7102–7110.

[118] Nils Lukas, Ahmed Salem, Robert Sim, Shruti Tople, Lukas Wutschitz, and Santiago Zanella-Béguelin. 2023. Analyzing leakage of personally identifiable information in language models. In *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 346–363.

[119] Lingjuan Lyu, Han Yu, and Qiang Yang. 2020. Threats to federated learning: A survey. *arXiv preprint arXiv:2003.02133* (2020).

[120] Lingjuan Lyu, Jiangshan Yu, Karthik Nandakumar, Yitong Li, Xingjun Ma, and Jiong Jin. 2019. Towards fair and decentralized privacy-preserving deep learning with blockchain. *arXiv preprint arXiv:1906.01167* 28 (2019).

[121] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. 2007. l-diversity: Privacy beyond k-anonymity. *Acm transactions on knowledge discovery from data (tkdd)* 1, 1 (2007), 3–es.

[122] Jose Maria Maestre, D Muñoz De La Peña, and Eduardo F Camacho. 2009. A distributed MPC scheme with low communication requirements. In *2009 American Control Conference*. IEEE, 2797–2802.

[123] N Boris Margolin, Brian N Levine, N Boris Margolin, and Brian Neil Levine. 2005. *Quantifying and discouraging sybil attacks*. Technical Report. Technical report, U. Mass. Amherst, Computer Science.

[124] Frank McKeen, Ilya Alexandrovich, Alex Berenzon, Carlos V Rozas, Hisham Shafi, Vedvyas Shanbhogue, and Uday R Savagaonkar. 2013. Innovative instructions and software model for isolated execution. *Hasp@ isca* 10, 1 (2013).

[125] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*. PMLR, 1273–1282.

[126] YJDA Miiller. 1990. Decentralized artificial intelligence. *Decentralised AI* 4, 1 (1990), 3–13.

[127] Fatemehsadat Mireshghallah, Huseyin A Inan, Marcello Hasegawa, Victor Rühle, Taylor Berg-Kirkpatrick, and Robert Sim. 2021. Privacy regularization: Joint privacy-utility optimization in language models. *arXiv preprint arXiv:2103.07567* (2021).

[128] Fan Mo, Hamed Haddadi, Kleomenis Katevas, Eduard Marin, Diego Perino, and Nicolas Kourtellis. 2021. PPFL: privacy-preserving federated learning with trusted execution environments. In *Proceedings of the 19th annual international conference on mobile systems, applications, and services*. 94–108.

[129] Payman Mohassel and Peter Rindal. 2018. ABY3: A mixed protocol framework for machine learning. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*. 35–52.

[130] Payman Mohassel and Yupeng Zhang. 2017. Secureml: A system for scalable privacy-preserving machine learning. In *2017 IEEE symposium on security and privacy (SP)*. IEEE, 19–38.

[131] Dilip Mookherjee. 2006. Decentralization, hierarchies, and incentives: A mechanism design perspective. *Journal of Economic Literature* 44, 2 (2006), 367–390.
[132] Satoshi Nakamoto et al. 2008. Bitcoin. *A peer-to-peer electronic cash system* 21260 (2008).
[133] Deepak Narayanan, Aaron Harlap, Amar Phanishayee, Vivek Seshadri, Nikhil R Devanur, Gregory R Ganger, Phillip B Gibbons, and Matei Zaharia. 2019. PipeDream: generalized pipeline parallelism for DNN training. In *Proceedings of the 27th ACM symposium on operating systems principles*. 1–15.
[134] Milad Nasr, Reza Shokri, and Amir Houmansadr. 2018. Machine learning with membership privacy using adversarial regularization. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*. 634–646.
[135] Seth Neel and Peter Chang. 2023. Privacy issues in large language models: A survey. *arXiv preprint arXiv:2312.06717* (2023).
[136] Thuy Dung Nguyen, Tuan Nguyen, Phi Le Nguyen, Hieu H Pham, Khoa D Doan, and Kok-Seng Wong. 2024. Backdoor attacks and defenses in federated learning: Survey, challenges and future research directions. *Engineering Applications of Artificial Intelligence* 127 (2024), 107166.
[137] Alexander Nilsson, Pegah Nikbakht Bideh, and Joakim Brorsson. 2020. A survey of published attacks on Intel SGX. *arXiv preprint arXiv:2006.13598* (2020).
[138] Pitch Patarasuk and Xin Yuan. 2009. Bandwidth optimal all-reduce algorithms for clusters of workstations. *J. Parallel and Distrib. Comput.* 69, 2 (2009), 117–124.
[139] Andrea Paudice, Luis Muñoz-González, and Emil C Lupu. 2019. Label sanitization against label flipping poisoning attacks. In *ECML PKDD 2018 Workshops: Nemesis 2018, UrbReas 2018, SoGood 2018, IWAISe 2018, and Green Data Mining 2018, Dublin, Ireland, September 10-14, 2018, Proceedings 18*. Springer, 5–15.
[140] Wenjun Peng, Jingwei Yi, Fangzhao Wu, Shangxi Wu, Bin Zhu, Lingjuan Lyu, Binxing Jiao, Tong Xu, Guangzhong Sun, and Xing Xie. 2023. Are you copying my model? protecting the copyright of large language models for eaas via backdoor watermark. *arXiv preprint arXiv:2305.10036* (2023).
[141] Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2020. Adapterfusion: Non-destructive task composition for transfer learning. *arXiv preprint arXiv:2005.00247* (2020).
[142] Krishna Pillutla, Sham M Kakade, and Zaid Harchaoui. 2022. Robust aggregation for federated learning. *IEEE Transactions on Signal Processing* 70 (2022), 1142–1154.
[143] Nikolaos Pitropakis, Emmanouil Panaousis, Thanassis Giannetsos, Eleftherios Anastasiadis, and George Loukas. 2019. A taxonomy and survey of attacks against machine learning. *Computer Science Review* 34 (2019), 100199.
[144] Richard Plant, Valerio Giuffrida, and Dimitra Gkatzia. 2022. You are what you write: Preserving privacy in the era of large language models. *arXiv preprint arXiv:2204.09391* (2022).
[145] Robert Podschwadt, Daniel Takabi, Peizhao Hu, Mohammad H Rafiei, and Zhipeng Cai. 2022. A survey of deep learning architectures for privacy-preserving machine learning with fully homomorphic encryption. *IEEE Access* 10 (2022), 117477–117500.
[146] Youyang Qu, Md Palash Uddin, Chenquan Gan, Yong Xiang, Longxiang Gao, and John Yearwood. 2022. Blockchain-enabled federated learning: A survey. *Comput. Surveys* 55, 4 (2022), 1–35.
[147] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.
[148] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems* 36 (2024).
[149] Jingjing Ren, Ashwin Rao, Martina Lindorfer, Arnaud Legout, and David Choffnes. 2016. Recon: Revealing and controlling pii leaks in mobile network traffic. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*. 361–374.
[150] M Sadegh Riazi, Mohammad Samragh, Hao Chen, Kim Laine, Kristin Lauter, and Farinaz Koushanfar. 2019. {XONN}:{XNOR-based} oblivious deep neural network inference. In *28th USENIX Security Symposium (USENIX Security 19)*. 1501–1518.
[151] Abhijit Guha Roy, Shayan Siddiqui, Sebastian Pölsterl, Nassir Navab, and Christian Wachinger. 2019. Braintorrent: A peer-to-peer environment for decentralized federated learning. *arXiv preprint arXiv:1905.06731* (2019).
[152] Mohamed Sabt, Mohammed Achemlal, and Abdelmadjid Bouabdallah. 2015. Trusted execution environment: What it is, and what it is not. In *2015 IEEE Trustcom/BigDataSE/Ispa*, Vol. 1. IEEE, 57–64.
[153] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. 2018. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *arXiv preprint arXiv:1806.01246* (2018).
[154] Zeyang Sha, Xinlei He, Pascal Berrang, Mathias Humbert, and Yang Zhang. 2022. Fine-tuning is all you need to mitigate backdoor attacks. *arXiv preprint arXiv:2212.09067* (2022).
[155] Yumeng Shao, Jun Li, Ming Ding, Kang Wei, Chuan Ma, Long Shi, Wen Chen, and Shi Jin. 2024. Design of Anti-Plagiarism Mechanisms in Decentralized Federated Learning. *IEEE Transactions on Services Computing* (2024).
[156] Virat Shejwalkar and Amir Houmansadr. 2021. Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In *NDSS*.
[157] Jiawen Shi, Yixin Liu, Pan Zhou, and Lichao Sun. 2023. Badgpt: Exploring security vulnerabilities of chatgpt via backdoor attacks to instructgpt. *arXiv preprint arXiv:2304.12298* (2023).
[158] Junyu Shi, Wei Wan, Shengshan Hu, Jianrong Lu, and Leo Yu Zhang. 2022. Challenges and approaches for mitigating byzantine attacks in federated learning. In *2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE, 139–146.

[159] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053* (2019).

[160] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*. IEEE, 3–18.

[161] Robin Staab, Mark Vero, Mislav Balunović, and Martin Vechev. 2023. Beyond memorization: Violating privacy via inference with large language models. *arXiv preprint arXiv:2310.07298* (2023).

[162] Heinrich von Stackelberg et al. 1952. Theory of the market economy. (1952).

[163] Sebastian U Stich. 2018. Local SGD converges fast and communicates little. *arXiv preprint arXiv:1805.09767* (2018).

[164] Latanya Sweeney. 2002. k-anonymity: A model for protecting privacy. *International journal of uncertainty, fuzziness and knowledge-based systems* 10, 05 (2002), 557–570.

[165] Ming Tang and Vincent WS Wong. 2021. An incentive mechanism for cross-silo federated learning: A public goods perspective. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 1–10.

[166] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805* (2023).

[167] Buse GA Tekgul, Yuxi Xia, Samuel Marchal, and N Asokan. 2021. Waffle: Watermarking in federated learning. In *2021 40th International Symposium on Reliable Distributed Systems (SRDS)*. IEEE, 310–320.

[168] Vale Tolpegin, Stacey Truex, Mehmet Emre Gursoy, and Ling Liu. 2020. Data poisoning attacks against federated learning systems. In *Computer Security–ESORICS 2020: 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, UK, September 14–18, 2020, Proceedings, Part I 25*. Springer, 480–501.

[169] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).

[170] Florian Tramer and Dan Boneh. 2018. Slalom: Fast, verifiable and private execution of neural networks in trusted hardware. *arXiv preprint arXiv:1806.03287* (2018).

[171] Florian Tramer and Dan Boneh. 2020. Differentially private learning needs better features (or much more data). *arXiv preprint arXiv:2011.11660* (2020).

[172] Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, Rui Zhang, and Yi Zhou. 2019. A hybrid approach to privacy-preserving federated learning. In *Proceedings of the 12th ACM workshop on artificial intelligence and security*. 1–11.

[173] Stacey Truex, Ling Liu, Mehmet Emre Gursoy, Lei Yu, and Wenqi Wei. 2019. Demystifying membership inference attacks in machine learning as a service. *IEEE transactions on services computing* 14, 6 (2019), 2073–2089.

[174] Xuezhen Tu, Kun Zhu, Nguyen Cong Luong, Dusit Niyato, Yang Zhang, and Juan Li. 2022. Incentive mechanisms for federated learning: From economic and game theoretic perspective. *IEEE transactions on cognitive communications and networking* 8, 3 (2022), 1566–1593.

[175] Joost Verbraeken, Matthijs Wolting, Jonathan Katzy, Jeroen Kloppenburg, Tim Verbelen, and Jan S Rellermeyer. 2020. A survey on distributed machine learning. *Acm computing surveys (csur)* 53, 2 (2020), 1–33.

[176] Pablo Villalobos, Jaime Sevilla, Lennart Heim, Tamay Besiroglu, Marius Hobbhahn, and Anson Ho. 2022. Will we run out of data? an analysis of the limits of scaling datasets in machine learning. *arXiv preprint arXiv:2211.04325* (2022).

[177] Danilo Vucetic, Mohammadreza Tayaranian, Maryam Ziaeefard, James J Clark, Brett H Meyer, and Warren J Gross. 2022. Efficient fine-tuning of BERT models on the edge. In *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 1838–1842.

[178] Sameer Wagh, Divya Gupta, and Nishanth Chandran. 2019. SecureNN: 3-party secure computation for neural network training. *Proceedings on Privacy Enhancing Technologies* (2019).

[179] Eric Wallace, Tony Z Zhao, Shi Feng, and Sameer Singh. 2020. Concealed data poisoning attacks on NLP models. *arXiv preprint arXiv:2010.12563* (2020).

[180] Alexander Wan, Eric Wallace, Sheng Shen, and Dan Klein. 2023. Poisoning language models during instruction tuning. In *International Conference on Machine Learning*. PMLR, 35413–35425.

[181] Hongyi Wang, Scott Sievert, Shengchao Liu, Zachary Charles, Dimitris Papailiopoulos, and Stephen Wright. 2018. Atomo: Communication-efficient learning via atomic sparsification. *Advances in neural information processing systems* 31 (2018).

[182] Jue Wang, Binhang Yuan, Luka Rimanic, Yongjun He, Tri Dao, Beidi Chen, Christopher Ré, and Ce Zhang. 2022. Fine-tuning language models over slow networks using activation quantization with guarantees. *Advances in Neural Information Processing Systems* 35 (2022), 19215–19230.

[183] Wenqi Wei, Ling Liu, Margaret Loper, Ka-Ho Chow, Mehmet Emre Gursoy, Stacey Truex, and Yanzhao Wu. 2020. A framework for evaluating client privacy leakages in federated learning. In *Computer Security–ESORICS 2020: 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, UK, September 14–18, 2020, Proceedings, Part I 25*. Springer, 545–566.

[184] Wenqi Wei, Ling Liu, Margaret Loper, Ka-Ho Chow, Mehmet Emre Gursoy, Stacey Truex, and Yanzhao Wu. 2020. A framework for evaluating client privacy leakages in federated learning. In *Computer Security–ESORICS 2020: 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, UK, September 14–18, 2020, Proceedings, Part I 25*. Springer, 545–566.

[185] Johannes Welbl, Amelia Glaese, Jonathan Uesato, Sumanth Dathathri, John Mellor, Lisa Anne Hendricks, Kirsty Anderson, Pushmeet Kohli, Ben Coppin, and Po-Sen Huang. 2021. Challenges in detoxifying language models. *arXiv preprint arXiv:2109.07445* (2021).

[186] Leon Witt, Mathis Heyer, Kentaroh Toyoda, Wojciech Samek, and Dan Li. 2022. Decentral and incentivized federated learning frameworks: A systematic literature review. *IEEE Internet of Things Journal* 10, 4 (2022), 3642–3663.

[187] Gavin Wood et al. 2014. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper* 151, 2014 (2014), 1–32.

[188] Yijia Xiao, Yiqiao Jin, Yushi Bai, Yue Wu, Xianjun Yang, Xiao Luo, Wenchao Yu, Xujiang Zhao, Yanchi Liu, Haifeng Chen, et al. 2023. Large language models can be good privacy protection learners. *arXiv preprint arXiv:2310.02469* (2023).

[189] Cong Xie, Sanmi Koyejo, and Indranil Gupta. 2019. Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance. In *International Conference on Machine Learning*. PMLR, 6893–6901.

[190] Yuan Xin, Zheng Li, Ning Yu, Michael Backes, and Yang Zhang. 2022. Membership leakage in pre-trained language models. (2022).

[191] Guowen Xu, Hongwei Li, Sen Liu, Kan Yang, and Xiaodong Lin. 2019. Verifynet: Secure and verifiable federated learning. *IEEE Transactions on Information Forensics and Security* 15 (2019), 911–926.

[192] Runhua Xu, Nathalie Baracaldo, and James Joshi. 2021. Privacy-preserving machine learning: Methods, challenges and directions. *arXiv preprint arXiv:2108.04417* (2021).

[193] Haomiao Yang, Kunlan Xiang, Mengyu Ge, Hongwei Li, Rongxing Lu, and Shui Yu. 2024. A comprehensive overview of backdoor attacks in large language models within communication networks. *IEEE Network* (2024).

[194] Qiang Yang, Anbu Huang, Lixin Fan, Chee Seng Chan, Jian Han Lim, Kam Woh Ng, Ding Sheng Ong, and Bowen Li. 2023. Federated Learning with Privacy-preserving and Model IP-right-protection. *Machine Intelligence Research* 20, 1 (2023), 19–37.

[195] Andrew C Yao. 1982. Protocols for secure computations. In *23rd annual symposium on foundations of computer science (sfcs 1982)*. IEEE, 160–164.

[196] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. 2018. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*. Pmlr, 5650–5659.

[197] Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, et al. 2021. Differentially private fine-tuning of language models. *arXiv preprint arXiv:2110.06500* (2021).

[198] Han Yu, Zelei Liu, Yang Liu, Tianjian Chen, Mingshu Cong, Xi Weng, Dusit Niyato, and Qiang Yang. 2020. A fairness-aware incentive scheme for federated learning. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. 393–399.

[199] Han Yu, Zelei Liu, Yang Liu, Tianjian Chen, Mingshu Cong, Xi Weng, Dusit Niyato, and Qiang Yang. 2020. A sustainable incentive scheme for federated learning. *IEEE Intelligent Systems* 35, 4 (2020), 58–69.

[200] Xixun Yu, Zheng Yan, and Athanasios V Vasilakos. 2017. A survey of verifiable computation. *Mobile Networks and Applications* 22 (2017), 438–453.

[201] Binhang Yuan, Yongjun He, Jared Davis, Tianyi Zhang, Tri Dao, Beidi Chen, Percy S Liang, Christopher Re, and Ce Zhang. 2022. Decentralized training of foundation models in heterogeneous environments. *Advances in Neural Information Processing Systems* 35 (2022), 25464–25477.

[202] Yufeng Zhan, Peng Li, Zhihao Qu, Deze Zeng, and Song Guo. 2020. A learning-based incentive mechanism for federated learning. *IEEE Internet of Things Journal* 7, 7 (2020), 6360–6368.

[203] Xinyang Zhang, Zheng Zhang, Shouling Ji, and Ting Wang. 2021. Trojaning language models for fun and profit. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 179–197.

[204] Lingchen Zhao, Qian Wang, Cong Wang, Qi Li, Chao Shen, and Bo Feng. 2021. Veriml: Enabling integrity assurances and fair payments for machine learning as a service. *IEEE Transactions on Parallel and Distributed Systems* 32, 10 (2021), 2524–2540.

[205] Shuai Zhao, Jinming Wen, Luu Anh Tuan, Junbo Zhao, and Jie Fu. 2023. Prompt as triggers for backdoor attack: Examining the vulnerability in language models. *arXiv preprint arXiv:2305.01219* (2023).

[206] Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou. 2016. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160* (2016).

[207] Ligeng Zhu, Zhijian Liu, and Song Han. 2019. Deep leakage from gradients. *Advances in neural information processing systems* 32 (2019).

[208] Michael Zhu and Suyog Gupta. 2017. To prune, or not to prune: exploring the efficacy of pruning for model compression. *arXiv preprint arXiv:1710.01878* (2017).

[209] Andy Zou, Zifan Wang, J Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043* (2023).