SoK: Private Knowledge Sharing in Distributed Learning

Yasas Supeksala
Swinburne University of Technology
Melbourne, Victoria, Australia

Thilina Ranbaduge
CSIRO's Data61
Canberra, ACT, Australia

Ming Ding
CSIRO's Data61
Sydney, NSW, Australia

Dinh C. Nguyen
University of Alabama
Huntsville, AL, USA

Bo Liu
University of Technology Sydney
Sydney, NSW, Australia

Caslon Chua
Swinburne University of Technology
Melbourne, Victoria, Australia

Jun Zhang
Swinburne University of Technology
Melbourne, Victoria, Australia

Abstract

The rapid advancement of Artificial Intelligence (AI) has transformed various industries, leading to the widespread distribution of AI models and data across intelligent systems. As modern data driven services increasingly integrate distributed knowledge entities, decentralized learning has become a prevalent approach to training AI models. However, this collaborative learning paradigm introduces significant security vulnerabilities and privacy challenges. This paper presents a comprehensive systematic review on private knowledge sharing in distributed learning, analyzing key knowledge components utilized in leading distributed learning architectures. We identify critical vulnerabilities associated with these components and examine defensive strategies to safeguard privacy while mitigating potential adversarial threats. Additionally, we highlight key limitations in knowledge sharing in distributed learning and propose future research directions to enhance security and efficiency in decentralized AI systems.

Keywords

Distributed learning, knowledge sharing, neural networks, artificial intelligence, knowledge components.

1 Introduction

Owing to the amplified sensitivity and enhanced computing capabilities of end user applications and Internet of Things (IoT) devices, an enormous amount of data is being generated and collected at an unprecedented pace. Traditional machine learning systems that integrate such data into a centralized device to process are becoming less feasible in real-world scenarios because of communication constraints and the computational overhead of large data silos. As a promising solution, *Distributed Learning* (DL) has garnered substantial attention in contemporary machine learning applications [160]. DL was introduced to enhance the efficiency, accuracy, and interpretability of distributed end nodes responsible for data collection and processing. By leveraging distributed learning, users can obtain more precise predictions with significantly lower computational overhead, making distributed learning more applicable to applications that involve distributed data [132].

DL has gained widespread adoption in industry [120]. However, for each end node (i.e. a data provider) in a DL framework, it is essential to enhance its processing power to enable DL. This requires the transition from single threaded algorithms to parallel algorithms [93]. We categorize distributed architectures into five main frameworks. Supervised, Unsupervised, Semi-supervised, Deep Reinforcement Learning, Distributed Transfer Learning, and Decentralized Large Language Models (DLLMs).

In Distributed Supervised Learning (DSL), the labeled data is stored across multiple nodes. These nodes collaboratively train models while maintaining data locality, enabling scalability and preserving privacy. Distributed Unsupervised Learning (DUSL) involves training on unlabeled data from distributed nodes. Distributed Semi-supervised Learning (DSSL) uses a small labeled dataset with a larger unlabeled one to make predictions. Distributed Reinforcement Learning (DRL) involves dividing experiences among agents, utilizing rewards for training without falling into the other three categories. It relies on environmental interaction to improve through trial and error. Distributed Transfer Learning (DTL) is best viewed as a hybrid architecture, as it combines the characteristics of a distributed learning architecture by involving decentralized computation and collaboration - with concepts from Federated Learning (FL), particularly privacy-preserving collaboration and decentralized model updates, while extending beyond FL by enabling knowledge transfer across heterogeneous tasks and domains.

Modern Large Language Models (LLMs), with parameter counts reaching into the hundreds of billions, introduce substantial challenges in scalability, data privacy, and computational efficiency. While conventional solutions primarily leverage parallelism within centralized infrastructure to manage these demands, our focus lies in distributed learning through decentralization. DLLMs address these concerns by enabling collaborative training or fine-tuning across multiple autonomous nodes or clients. In this decentralized setup, participants keep their local data and collaboratively update a global model without sharing raw information, enabling privacy-preserving learning in distributed environments.

Proceedings on Privacy Enhancing Technologies 2025(4), 485–506

https://doi.org/10.56553/popets-2025-0141

1.1 Distributed Learning vs. Federated Learning

Federated Learning (FL) architectures, although related, are not easily classified as traditional distributed learning (DL) due to their distinct characteristics, particularly the independence of participants and their federated network structure. In DL, local devices train on their own data and frequently synchronize with others or a central server by exchanging partial model updates throughout the training process [101]. In contrast, FL performs full local training, sharing only aggregated model parameters periodically, which helps preserve data privacy. DL often involves sharing various forms of intermediate information, such as features, predictions, or processed data, which can increase bandwidth demands and privacy risks. FL, in contrast, restricts communication to model updates, making it more suitable for privacy-sensitive scenarios where data cannot leave local devices. Additionally, DL typically operates in high-bandwidth, reliable environments such as data centers. However, FL must deal with unreliable and heterogeneous clients, which introduce challenges to consistency and convergence.

1.2 Distributed Learning Based Modern Applications

To understand modern DL applications in a comprehensive way, it is essential to explore key frameworks such as Apache Spark, Apache Flink and TensorFlow Distributed, each offering unique capabilities in scalability, real-time analytics, and distributed machine learning.

Apache Spark [140], known for its resilient distributed data sets (RDDs) and in-memory processing, provides high-performance solutions for big data analytics, complex batch processing, ETL (Extract, Transform, Load) operations, and streaming analytics. Its integration with high-level APIs, such as PySpark, SparkSQL, and Spark Streaming, enables efficient management of massive datasets in distributed environments, significantly reducing latency and supporting iterative computations in DL contexts.

The Apache Flink stream-first architecture and precise state management make it ideal for real-time analytics and event-driven applications [111]. Its event-driven runtime supports continuous streaming data, enabling stateful computations and accurate event-time processing. Flink's windowing and checkpointing ensure fault tolerance and consistency, which are key to continuous learning tasks in dynamic environments such as IoT sensor data or user behavior analytics.

TensorFlow Distributed offers strong support for distributed deep learning and large-scale model training via data and model parallelism [135]. Through synchronous and asynchronous gradient descent, it optimizes resource use across heterogeneous environments, including multi-GPU and multi-node clusters. Libraries like TensorFlow Extended (TFX) streamline the deployment pipeline, supporting end-to-end workflows from data ingestion to model monitoring.

Using these tools, DL systems achieve faster processing, greater scalability, and efficient handling of diverse workloads, from real-time analytics to complex model training. In DL frameworks, reliable communication between nodes is essential for model collaboration, interactive learning, and consistent results [21]. These communication protocols, shaped by architectural designs and application needs [9], often involve the exchange of critical components, such as data batches or model output. We define these as the *knowledge components*, including parameters that influence the predictions or residual artifacts from training.

However, sharing such data and parameters between nodes introduces cybersecurity risks, such as data leakage, inference attacks, and unauthorized parameter reconstruction. These risks are mitigated through privacy-preserving techniques such as Homomorphic Encryption (HE) [40] and Differential Privacy (DP) [177]. HE allows secure computation on encrypted data, while DP protects privacy by adding calibrated noise to data exchanges, minimizing risks to individual data points.

1.3 Contributions of This SoK

This systematic review aims to provide a comprehensive view on knowledge sharing in DL. Focus on the parameters shared during the communication rounds and the vulnerabilities associated with them. Later we highlight the limitations associated with the existing architectures and provide future directions. In summary, our contribution can be summarized as follows:

(1) Our investigation of key DL architectures includes a categorization of their strategies and knowledge components, along with a comparative summary in Table 1.
(2) We identify and discuss critical vulnerabilities associated with each knowledge component, assuming that the communication phase in distributed learning is prone to attacks.
(3) In the recent literature, we explore different attacks and methodologies that can exploit different knowledge components. To clarify the rationale for selecting attack and defense mechanisms in our analysis of privacy risks in distributed knowledge sharing, we developed a table of selection criteria. Table 7 summarizes the rationale for the inclusion of each attack and defense based on established literature.
(4) We examine the application of DL in LLMs, focusing on the implementation of DL, addressing privacy concerns, and exploring privacy-preserving techniques.
(5) Finally, we discuss the limitations of current DL architectures, particularly their susceptibility to vulnerabilities in distributed systems. We then explore robust defense mechanisms that can mitigate these vulnerabilities. Table 1 summarizes our contributions in comparison to other surveys presented in the literature.

1.4 Literature Selection Criteria

We developed our selection criteria through a systematic review of the literature exploring the intersection of knowledge sharing, architectural design, and privacy in DL systems. To enable a rigorous analysis of privacy risks and mitigation strategies, we introduced a qualitative framework that maps the relationships between knowledge components, attack vectors, and defense mechanisms. This framework is represented through structured tables in Appendix A Table 7 across five DL paradigms: DSL, DUSL, DSSL, DRL, and DTL. While decentralized LLMs were initially considered in our taxonomy, they were excluded from the selection criteria due to limited empirical evidence and the absence of standard threat models and benchmarks. Our tables were constructed through an interpretive analysis of the types of knowledge exchanged between nodes, such

as gradients, logits, embeddings, and internal states, to identify potential vulnerabilities. Each knowledge component was assessed for its degree of exposure to various attacks using a qualitative scale based on three key factors:

- Whether the knowledge is directly exploitable by an attacker.
- The difficulty of obtaining or reverse engineering the *knowledge*.
- The success likelihood of an attack based on prior empirical studies.

For instance, in the case of Gradient Leakage (GL) attacks, gradients were classified as highly exposed because they have been shown to allow effective reconstruction of training data, especially in early learning phases. We evaluated the impact of an attack aligning the required attack inputs with the knowledge actually shared in each architecture, using literature from 2019 to 2025. In parallel, defense mechanisms were evaluated on the basis of whether they obscure, limit, or encrypt the targeted knowledge, such as homomorphic encryption protecting gradient computations or Differential Privacy defending against inference-based threats. Appendix A Table 7 complements this analysis by summarizing the required inputs of each attack, the vulnerable model types, and the corresponding defensive strategies. This table offers a structured and comparative view of the evolving threat landscape in DL. For clarity and ease of reference, Appendix B Table 8 provides a complete list of abbreviations and technical terms used throughout this paper.

2 Knowledge Sharing in a Distributed Learning Setting

DL is a powerful method that combines the computational resources and data of multiple nodes to achieve high precision, enabling high-performance machine learning models to handle large amounts of data [160]. By leveraging the collective power of multiple participants, DL supports sophisticated decision making in intelligent systems increasingly reliant on artificial intelligence.

DL can be regarded as a form of *parallel learning*, where traditional single-thread algorithms are transformed into parallel systems [70]. Two main approaches to parallelism in DL are (1) *data parallelism* and (2) *model parallelism*. Although data parallelism has been well studied, this systematic review focuses on model parallelism or model decentralization, which examines how *knowledge* is generated through machine learning models. This is crucial as machine learning models identify *unique patterns* and make *informed decisions*.

Model parallelism isolates knowledge components during DL and can be categorized into three architectures [179]:

- *Tensor parallelism*: Distributes *model weights*, *gradients*, and *optimizer states* across devices to facilitate distributed forward and backward propagation.
- *Layer wise parallelism*: Also known as *optimizer state sharing*, it employs a replica of a single optimizer in parallel data ranks.
- *Layer pipelining*: Partitions weights while preserving their integrity, utilizing either *synchronous* or *asynchronous pipeline parallelism* to support efficient hardware utilization.

In Sections 3 - 7 we analyze distributed deep learning architectures that use model parallelism and identify critical *shared information*, referred to as *knowledge*. We identify more than twenty *knowledge components* and categorize them into four main groups for better management.

- *Neural Network Update Information*: Knowledge components include *gradients*, *weights*, *batch size*, and *object size*, shared during iterative updates.
- *Neural Network Output Information*: Shared output information includes *logits*, *layer data* (e.g.attention and output layers), and *immediate partition outputs*.
- *Neural Network Parameter Information*: Components include *parameter distribution*, *aggregation parameters*, *skewness factors*, *tangents of data manifolds*, *partitioning points*, and *control parameters*.
- *Neural Network Reinforcement Action Information*: Includes components such as *cell state*, *memory*, *latent distribution mean and variance*, *policy gradients*, and *rewards*, which facilitate reinforcement learning.

The subsequent sections explore knowledge components in various architectures, evaluate vulnerabilities to attacks targeting knowledge sharing , and discuss defensive mechanisms to address these challenges.

3 Knowledge Sharing in Distributed Supervised Learning

Supervised learning plays an important role when considering *Knowledge sharing among AI*. The use of supervised learning is prevalent in modern AI applications and this approach can also be implemented in a distributed environment. To gain a deeper understanding of the knowledge elements involved in the collaborative learning stage, we have broken down DSL into three main architectural types. We will delve into how these traditional machine learning architectures are utilized in deep learning and examine the knowledge components associated with each one in the following sections.

3.1 Distributed Supervised Learning Architectures

3.1.1 Distributed Training for Multi Layer Perceptrons. A Multilayer Perceptron (MLP) is a classic supervised learning model composed of an input layer, hidden layers, and an output layer [45]. In distributed settings, the model leverages multiple devices to handle larger datasets and more complex learning tasks. Data is typically partitioned across these devices, each training a portion of the network, and the collective outputs are then aggregated into a final model. Within such architectures, studies like Chang et al. [19] introduced *DeepLinQ*, a blockchain based privacy aware MLP design that incorporates distributed optimization and hardware aware techniques. Similarly, Xia et al. [176] highlighted communication challenges, proposing mechanisms that allow training to continue despite packet losses. Their approach emphasizes parameter distribution and aggregation parameters to maintain effective learning without strictly relying on synchronous gradient updates.

Paper	Att & Def	Learning Architecture				Discussion on Knowledge Sharing	Shared Knowledge
		DSL	DUSL	DSSL	DDRL		
[160]	✗	✓	✓	✓	✓	Touches knowledge sharing based on the communication phase.	Discusses the implications of distributed systems over conventional ML with challenges and limitations.
[75]	✓	✗	✗	✗	✓	Discusses knowledge sharing based on communication efficiency and model convergence.	A comprehensive survey on DL trade-offs in communication networks.
[141]	✓	✗	✗	✗	✓	Explores how model participants share knowledge.	Survey on strategies for adapting DL to edge and fog computing.
[39]	✗	✓	✗	✗	✗	Discusses edge intelligence approaches but does not specify knowledge components.	Investigates challenges of running ML models at the network edge in a distributed manner.
[190]	✓	✓	✓	✗	✓	Examines classification algorithms in a distributed setting but does not focus on specific knowledge components.	Survey on privacy concerns in centralized patient data and alternative distributed processing approaches.
[143]	✗	✓	✗	✗	✗	Explores ML aggregation in DL but lacks focus on specific architectures.	Investigates data aggregation techniques in distributed ML.
[32]	✓	✓	✓	✗	✗	Considers knowledge sharing in infrastructure optimization.	Survey on resource-aware device placement in distributed edge networks.
Our Work	✓	✓	✓	✓	✓	Investigates distributed ML architectures and shared knowledge components.	Identifies vulnerabilities, exploits, and privacy-preserving techniques.

Table 1. Comparison of Surveys of Knowledge Sharing in Distributed Learning Architectures. We categorize papers based on attacks and defenses (Att & Def), learning architectures, Distributed Supervised Learning (DSL), Distributed Unsupervised Learning (DUSL), Distributed Semi supervised Learning (DSSL), and Distributed Deep Reinforcement Learning (DDRL).

3.1.2 Distributed Convolutional Neural Networks. Convolutional Neural Networks (CNNs) specialize in tasks such as image recognition and analysis [34]. In distributed CNNs, major layers like convolution and pooling are partitioned to different nodes for parallelization [18]. Boulila et al. [15] proposed a two step process involving big data ingestion and splitting satellite images for supervised classification, followed by a distributed CNN for final classification. Stahl et al. [150] showed how layer fusion and partitioning can optimize resource usage, while Zhang et al. [199] presented *AD-CNN*, which dynamically assigns tasks (e.g., convolution/pooling layers) to edge nodes based on real time operational status. These strategies help handle larger image datasets, reduce training time, and maintain scalability.

3.1.3 Distributed Recurrent Neural Networks. Recurrent Neural Networks (RNNs), including Long Short Term Memory (LSTM) and Gated Recurrent Units (GRU) variants, process sequential data by retaining information from previous states [189]. In distributed RNNs, local models train on subsets of data and share parameters via consensus strategies to achieve a global optimum [129]. Distributed LSTM approaches frequently appear in language modeling and machine translation, focusing on methods such as mini batch distribution [30, 139]. Meanwhile, distributed GRUs can integrate FL frameworks for cybersecurity applications [152].

3.2 Knowledge Components in Distributed Supervised Learning

In *Gradients and Parameter Distributions (MLPs)* Chang et al. [19] demonstrated that *gradients* shared among siloed nodes represent a core knowledge element. Xia et al. [176] illustrated how *parameter distribution* and *aggregation parameters* can facilitate learning while mitigating delays due to packet loss. These distributions and aggregations act as the outcomes of multiple learning iterations and are therefore considered knowledge.

In *Layer Outputs and Partitioning (CNNs)*, the *convolution* and *pooling* layers are often distributed, while the fully connected layer may remain on a central node [150]. The specific partitioning strategies (*partitioning points*) and decisions on how to allocate these layers (the *skewness factor*) function as shared knowledge in collaborative training [127]. In distributed LSTM, *cell states*, *hidden cell states*, and selected *weights* (e.g., forget gate parameters) are propagated among nodes [105]. For distributed GRUs, *logits* and *gradients* frequently form the primary knowledge elements to be exchanged [152].

These knowledge components (gradients, parameter distributions, layer outputs, and states) guide the collaborative processes in distributed MLP, CNN, and RNN architectures. They are essential for efficient training and remain targets for privacy and security considerations, as discussed in Section 3.3.

3.3 Attacks and Defenses in Distributed Supervised Learning

Distributed MLPs are susceptible to a range of attacks that exploit core knowledge components such as gradients, shared model parameters, and aggregation mechanisms. Model memorization attacks [153] can target these components in both white-box and black-box scenarios, allowing adversaries to reconstruct or manipulate sensitive information. White-box attacks use internal model details to extract data, while black-box attacks infer information from outputs alone.

Knowledge Component	Attacks										Defenses					
	MM	GL	MI	PL	PU	DS	FE	IA	DP	MV	HE	DP	SA	EA	OB	TEE
Gradients [152, 176, 179]	◕	●	◑	○	◕	◑	◑	◕	●	◕	●	◕	●	●	◑	◕
Parameter Distribution [55, 176]	◕	●	◕	◑	●	◕	◕	●	●	◕	●	◕	●	●	◑	◕
Aggregators [55, 176]	◑	●	◕	◑	●	◕	●	●	◕	○	●	◕	●	●	◑	●
Convolution Layer [34, 150, 199]	◕	●	◕	◑	●	◕	●	●	●	●	●	◕	◑	●	○	○
Pooling Layer [34, 199]	◑	◕	◑	◑	◕	◑	◕	◕	◕	◑	●	◕	●	○	○	○
Fully Connected Layer [150, 199]	◕	●	◕	◑	●	◕	◕	◕	●	◕	●	◕	◕	●	○	○
Cell States (LSTM) [105]	◕	●	◕	◑	●	◕	◕	◕	●	◕	◕	◑	◕	○	○	○
Logits (GRU) [152]	◑	◕	◕	◑	●	◕	◑	◕	●	◑	◕	◑	◕	●	◑	◕
Skewness Factor [127, 150]	◕	●	◕	◑	●	◕	◕	●	●	◕	◕	◕	◑	◕	◔	◕
Partitioning Points [127, 150, 199]	◕	◕	●	◑	●	●	●	◕	●	◕	◕	◑	●	●	○	◕

Table 2. Attack types: Model Memorization (MM), Gradient Leakage (GL), Membership Inference (MI), Packet Loss (PL), Malicious Parameter Updates (PU), Data Skewing (DS), Feature Estimation (FE), Inference Attacks (IA), Data Poisoning (DP), and Model Inversion (MV). Defensive measures: Homomorphic Encryption (HE), Differential Privacy (DP), Secure Aggregation (SA), Encryption-based Secure Aggregation (EA), Obfuscation (OB), and Trusted Execution Environments (TEE). Vulnerability levels of knowledge components (● - critically exposed, ◕ - significant risk, ◑ - attack is possible but harder to exploit, ◔ - minor attack surface, ○ - mostly resistant) and effectiveness of defensive strategies (● - highly effective, ◕ - significantly reduces risk, ◑ - provides partial protection, ◔ - limited effectiveness, ○ - minimal protection).

In distributed CNNs, vulnerabilities arise at layer boundaries, data partitioning schemes, and due to non-i.i.d. (skewed) data distributions. Attacks such as model inversion [172], data skewing [76] and membership inference attacks [147] exploit these characteristics to reconstruct training data or influence model behavior.

To mitigate these risks, defenses like obfuscation (e.g., randomized activation masking, noise injection) [182] and Trusted Execution Environments (TEEs) [123] have been proposed. However, these mechanisms often trade off model accuracy, training efficiency, and system complexity, highlighting the need for careful integration into distributed learning pipelines. A comprehensive mapping of shared knowledge components, associated attack surfaces, and corresponding defense mechanisms is provided in Table 2.

Our systematic review identifies core privacy risks in DSL stemming from knowledge sharing across model architectures.

- MLPs are vulnerable to model memorization and gradient leakage due to their dense connections and straightforward computations.
- CNNs exhibit leakage risks amplified by partitioning strategies and data skewness, particularly in heterogeneous settings. Defenses such as differential privacy and obfuscation are effective mitigations.
- LSTM and GRU models introduce temporal risks through shared cell states and sequential gradients. While federated learning helps decentralize exposure, it also introduces communication and convergence challenges.

4 Knowledge Sharing in Distributed Unsupervised Learning

Unsupervised learning is a technique for finding patterns in unlabeled data without human direction. Advanced unsupervised algorithms can uncover hidden patterns in massive datasets, improving accuracy and reliability for classification and prediction tasks. These algorithms examine unlabeled input data that have not been organized into specific categories. Instead, they don't have a predefined output and focus on discovering relationships and patterns within the input data. The training of the machine learning model utilizes unlabeled input data. Initially, it interprets the raw data to uncover hidden patterns. Subsequently, k-means clustering and other suitable algorithms are applied to group data objects based on their similarities and dissimilarities [102].

Unsupervised machine learning algorithms are mainly divided into clustering, which identifies inherent groupings, and association, which discovers specific rules based on the application's needs. Additionally, these algorithms can perform dimensionality reduction tasks using methods like Principal Component Analysis (PCA) and autoencoders. The architecture is defined by selecting an algorithm that groups data based on similarities and differences. When these algorithms are deployed in a distributed environment, it is termed *Distributed Unsupervised Learning* (DUSL), enhancing effectiveness, accuracy, and scalability for large datasets. This *DUSL architecture* is categorized into *distributed generative adversarial networks*, *distributed autoencoders*, and *distributed self organizing maps*, with each category featuring a unique *knowledge sharing* scheme that presents distinct vulnerabilities and defense mechanisms.

4.1 Distributed Unsupervised Learning Architectures

4.1.1 Distributed Generative Adversarial Networks. Generative Adversarial Networks (GANs) combine a generator and a discriminator to learn from data in a two player game [25]. In a distributed setting, GAN training shifts to the edge, leveraging data parallelism or hybrid (data+model) parallelism. One approach is to place multiple generators at edge nodes and use a single global discriminator [162]. Another method, PATE-GAN [80], integrates the Private Aggregation of Teacher Ensemble framework to preserve privacy by leveraging multiple teacher discriminators and a student discriminator.

MG-GAN [27] extends the idea of multi-generator GANs for pedestrian trajectory prediction, reducing out-of-distribution samples by using specialized generators. Discriminator-based GANs like Hardy et al. [59] adopt a single generator on a parameter server, with discriminators exchanging peer-to-peer updates. Multi-generator architectures address mode collapse via hierarchical layers, distributing generators and discriminators to maintain variety in synthetic data. Reinforcement learning can also be integrated for dynamic controller parameters, as seen in [146].

4.1.2 Distributed Autoencoders. Autoencoders are widely used for feature selection and dimensional reduction [58]. They typically consist of an encoder mapping input data to a latent representation and a decoder reconstructing the input from this representation. In distributed autoencoders, the encoder often resides at multiple nodes, with the decoder centralized in a global model [97, 202]. Mechanisms like DRASTIC [29] distribute recurrent encoders to a single collaborative decoder, ensuring model performance comparable to a monolithic version trained on all data. Some architectures distribute both encoder and the decoder for improved utility, as in [104], where spatial pattern recognition benefits from encoding messages into cell seeding configurations.

4.1.3 Distributed SOM. Self Organizing Maps (SOMs) use competitive learning to cluster and reduce dimensionality of high dimensional data [88]. To handle large scale distributed data, variants like DSOM split computations among nodes [133], improving scalability and speed. Some implementations incorporate FL to avoid centralizing data [84]. Distributing SOMs across multiple nodes can also mitigate denial-of-service (DoS) risks [85]. Regardless of specific application, distributed SOMs initialize and train local maps on local data, then merge them (a weighted sum) into a global SOM to ensure consistent clustering and classification.

4.2 Knowledge Components in Distributed Unsupervised Learning

In these distributed unsupervised architectures, *knowledge* emerges through shared outputs, updated parameters, or specific statistical components. *In GANs*, knowledge components include *learning iterations*, *error feedback*, and *message size* [59]. In multi-generator setups (e.g., MG-GAN [27]), *logits* serve as the primary outputs from specialized generators. Approaches like PATE-GAN [80] introduce *control parameters* and *votes* from teacher discriminators to a student discriminator. This distributed discriminator architecture is depicted in Figure 1. Meanwhile, hierarchical methods

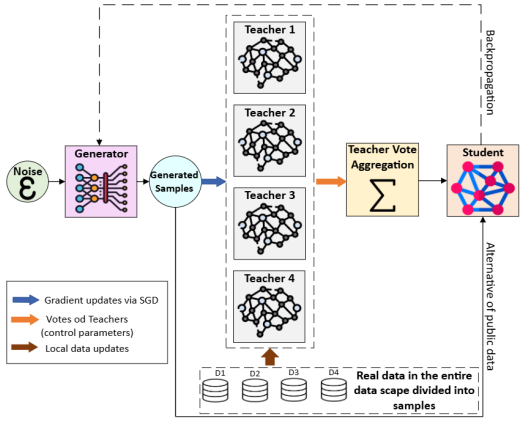

Figure 1. The Learning architecture of DL through distributed discriminators.

(M-GAN [72]) leverage *parameter distributions* beyond input layers to tackle mode collapse.

In Autoencoders, knowledge typically manifests in the *encoded representations* or *latent distributions* sent from distributed encoders to a global or collaborative decoder [29]. Semi-supervised settings add *latent distribution mean* and *variance* statistics for classification [104]. SOMs use the final global map generated from merging locally trained SOMs [133]. Thus, *merged classification logits* represent a significant knowledge component, as they ensure consistent and accurate representation of the distributed data.

These knowledge components, ranging from *gradients*, *logits*, and *parameter distributions* to *latent statistics*, are central to collaborative learning in distributed systems but also pose privacy and security risks.

4.3 Attacks and Defenses on DUSL

Distributed Unsupervised Learning (DUSL) involves the exchange of knowledge components such as *logits*, *weights*, *controller parameters*, and *gradients*, rendering it susceptible to various security threats, including *feature estimation*, *model reconstruction*, and *model poisoning*. These vulnerabilities primarily emerge during communication phases, where secondary attributes such as *batch sizes* and *distribution means* may be intercepted and exploited by adversaries.

An adversary can, for instance, perform *feature estimation* by monitoring gradient updates or partial logits to approximate the underlying data distribution. In *model reconstruction* attacks, the adversary uses leaked information (e.g., weights, partial gradients) to recreate or approximate the global model, thereby gaining insight into private data sets. *Model poisoning* represents a more aggressive scenario, in which adversaries inject malicious updates designed to degrade the overall performance of the model or introduce targeted biases.

One avenue of defense is *gradient sharing reduction* . By restricting the frequency or granularity of gradient exchange, it becomes harder for an attacker to infer sensitive data. Techniques may involve quantizing or masking gradients before transmission to reduce the potential for reconstruction [62]. Another is *differential*

Knowledge Component	Attacks										Defenses			
	MI	MP	PI	PR	EA	PP	IA	MR	FE	IM	DP	DR	EN	DA
Logits (GANs, SOMs) [27, 198]	◕	◔	●	◕	◑	◔	◕	◕	●	◑	◕	◑	●	◑
Gradients [146, 198]	◑	●	◕	◕	◑	◔	◕	●	◕	○	●	◕	◕	○
Controller Parameters (GANs) [72, 80, 146]	◕	◕	◑	◕	◔	◕	●	◕	◑	○	◕	◑	◕	◑
Batch Sizes [59]	○	◔	○	◑	○	○	◑	◔	○	○	◑	○	○	●
Error Feedback (GANs) [59]	◕	◕	◑	◕	◑	◑	◕	◕	●	○	◕	◑	◑	◑
Latent Dist. Mean/Var. (Autoencoders) [29, 104]	◕	◑	●	●	◔	◑	◕	◕	●	○	◕	●	◕	◕
Data Rep. Snippets (Autoencoders) [29]	●	◑	●	●	◕	◑	◕	●	◕	○	●	◕	◑	●
Merged Class. Logits (SOMs) [84, 85, 133]	●	◕	●	◕	◕	◔	◕	●	◕	●	◕	◑	◕	◕

Table 3. Attack types: Membership Inference (MI), Model Poisoning (MP), Property Inference (PI), Privacy Re-identification (PR), Eavesdropping Attacks (EA), Preference Poisoning (PP), Inference Attacks (IA), Model Reconstruction (MR), Feature Estimation (FE), and Interception during Map Merging (IM). Defensive measures: Differential Privacy (DP), Dimensionality Reduction (DR), Encryption (EN), and Data Augmentation (DA). Vulnerability levels of knowledge components (● - critically exposed, ◕ - significant risk, ◑ - attack is possible but harder to exploit, ◔ - minor attack surface, ○ - mostly resistant) and effectiveness of defensive strategies (● - highly effective, ◕ - significantly reduces risk, ◑ - provides partial protection, ◔ - limited effectiveness, ○ - minimal protection).

privacy (DP), where calibrated noise is added to the updates before sharing, offering formal privacy guarantees but potentially impacting the accuracy of the model [62]. DP introduces calibrated noise into training data, gradients, or outputs to prevent the leakage of sensitive information [1]. By bounding the influence of any single data point, DP enables formal privacy guarantees even under adversarial inference, though it often incurs a trade-off in model accuracy and convergence speed. *Dimensionality Reduction (DR)* techniques such as PCA or autoencoders can further minimize privacy risks by projecting sensitive data into lower-dimensional spaces, reducing the exposure of fine-grained patterns exploitable by attackers.

In parallel, *Encryption (EN)* methods, ranging from symmetric encryption to fully homomorphic encryption, secure data and model parameters during transmission and computation [4]. Although encryption ensures confidentiality against passive adversaries, performance overhead remains a limiting factor, particularly in real-time or resource-constrained environments.

Finally, *Data Augmentation (DA)* can enhance privacy and model robustness by synthetically expanding the training set, making it harder for attackers to link model behaviors to specific inputs [186]. Techniques such as mixup, random cropping, or adversarial augmentation introduce controlled variability that masks original data distributions without requiring architectural changes.

Balancing these defenses in distributed settings such as DUSL remains a challenge. Each technique offers distinct trade-offs across privacy, utility, and efficiency dimensions. Hence, adaptive integration strategies, guided by task requirements and threat models, are essential to achieve practical and secure learning frameworks. A comprehensive overview of the knowledge components, associated attack vectors, and corresponding defense strategies is provided in Table 3.

This section explores knowledge sharing in DUSL, focusing on architectures like GANs, Autoencoders, and SOMs.

- We have identified unique vulnerabilities tied to shared components such as logits and latent representations. These components expose models to property inference, model reconstruction, and poisoning attacks.
- These attacks are particularly difficult to detect in unsupervised settings due to the lack of labeled data. Without ground truth, anomalies in latent structures or clustering outputs can go unnoticed, and implicit feature manipulations may not manifest in obvious output errors.
- Defense mechanisms like DP and encryption effectively reduce information leakage but come with trade-offs in computation, scalability, and model utility. Emerging strategies such as partition obfuscation offer a promising balance between privacy and performance.

5 Knowledge Sharing in Distributed Semi-supervised Learning

Semi-supervised learning addresses the challenge of classifying large, heterogeneous, and difficult to label data commonly found in modern edge and cloud device services. This method is widely used in industry to process vast amounts of data obtained from smart devices like IoT sensors, which are abundant but hard to classify due to their diversity. In current applications, semi-supervised learning is a key approach for handling such data [211].

This semi-supervised learning architecture combines classification and clustering algorithms to group data based on similarity and then uses clustering algorithms to determine the relevance of the data samples. The data can then be labeled and used for machine learning. It is clear that semi-supervised learning can be used

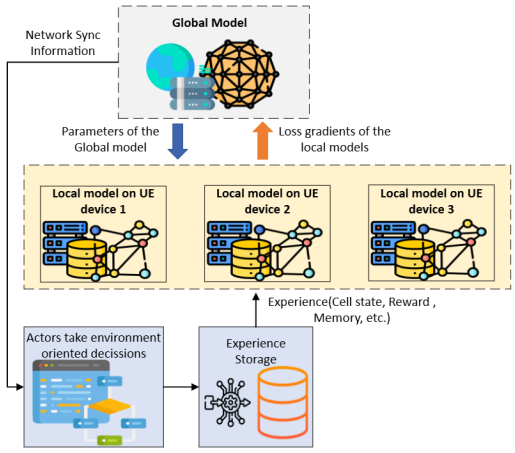

Figure 2. Knowledge Sharing in Distributed Semi-supervised Learning.

for both inductive and transductive learning tasks [161]. Based on these approaches, we recognize three major architectures of semi-supervised learning: *semi-supervised GANS*, *distributed transformers*, and *distributed contrastive learning* mechanisms. The distributed semi-supervised learning (DSSL) process, depicted in Figure 2, forms the foundation for the architectures discussed in Sections 5.1.1, 5.1.2, and 5.1.3.

5.1 Distributed Semi-supervised Architectures

5.1.1 GANs. Semi-supervised GANs expand the functionality of conventional GANs (originally unsupervised) by incorporating a supervised component to handle limited labeled data alongside a larger pool of unlabeled data [197]. In a *distributed* setting, this architecture is referred to as *Distributed Semi-supervised GANs (DSSGAN)*.

Zhou et al. [209] proposed generating additional data for minority classes to address imbalanced datasets, effectively leveraging GANs in a semi-supervised manner. Similarly, [68] expands this to fully utilize labeled and unlabeled data for robust classification. Once deployed in a distributed environment, where generators, discriminators, and classification modules can be decentralized across multiple nodes, these setups form a DSSGAN. As shown in Figure 2, the independence of these components underscores the DL paradigm.

5.1.2 Distributed Transformers. Transformers [159] utilize self attention mechanisms for sequence-to-sequence tasks in natual language processing (NLP) and related fields. When distributed, transformers handle large datasets efficiently. FeSTA [130], for instance, applies split learning with vision transformers [31] to achieve distributed training, whereas [127] introduces an *Agnostic ViT* approach, distributing a transformer as a central model and offloading specialized tasks (e.g., convolutional heads) to client nodes. The Pipe Transformer [64] further optimizes distributed training with an elastic pipelining process. These methods exemplify model parallelism and data parallelism in distributed transformers to handle large scale tasks efficiently.

5.1.3 Distributed Contrastive Learning. Contrastive learning [57] is a semi-supervised paradigm that learns a generalized data representation by comparing augmented views of the same or different samples. In a distributed framework, Federated Contrastive Learning (FCL) [174] applies this idea to medical images, sharing encoded features between multiple parties while preserving privacy. This decentralized approach uses contrastive loss to improve performance, as each client contributes to a more diverse dataset through locally computed representations. Despite its relative novelty, FCL [174] stands out as a state-of-the-art solution that integrates semi-supervised contrast learning in a distributed environment.

5.2 Knowledge Components in Distributed Semi-supervised Learning

Across these DSSL architectures (DSSGAN, Transformers, and Contrastive Learning), core *knowledge components* can be found, DSSGAN blends traditional unsupervised GAN elements (generator-discriminator interplay) with a supervised classifier. Key *knowledge* often includes the *tangents of the data manifold* estimated by the generator [91, 209], as well as *gradients* and *parameters* for classification tasks [86]. Distributed classifiers can also exchange *logits* to fine tune decision boundaries [54]. In distributed transformer architectures, communication revolves around *task specific head/tail parameters*, *gradients*, and selected outputs of crucial layers (e.g., the last multi headed attention). Mechanisms like permutations or pipelining protect data confidentiality [128, 130] while sharing these intermediate outputs and gradients among nodes.

Distributed Contrastive Learning frameworks (DCL) [174] focus on exchanging *continuous predictive logits* and *pretext logits* related to augmented samples. In both the pretext and supervised phases, the *intermediate data representation* forms a central *knowledge* element. When multiple target models are appended, *gradients* and *parameters* associated with each client's local updates similarly become essential knowledge sharing components. By identifying these knowledge components across DSSGAN, distributed transformers, and DCL, one can better understand how semi-supervised tasks are tackled collaboratively while still addressing privacy and security concerns inherent to distributed machine learning.

5.3 Attacks and Defenses on Distributed Semi-supervised Learning

DSSL continues to face substantial security challenges due to the interplay between a small labeled dataset and a vast pool of unlabeled data. As *knowledge* is exchanged during the supervised phase, vulnerabilities arise when *data representations* are shared or exposed.

For instance, DSSGANs rely on *tangents of the data manifold* [91, 209] to guide generator training, which can be intercepted by adversaries aiming to perform *data poisoning* [16, 155] or *model reconstruction* [109, 204]. Furthermore, *gradients* [54, 86, 130] and *parameters* [127, 128, 159] shared in DSSGAN, Transformer, and Contrastive Learning frameworks are susceptible to *gradient leakage* [155] and *parameter inference* [109].

Knowledge Component	Attacks									Defenses			
	DP	MR	GL	PI	DI	GR	MP	FE	AA	WS	OB	PS	SR
Tangents of Data Manifold (DSSGANs) [91, 209]	○	○	○	○	◑	○	◕	◑	◕	◑	◕	◑	◕
Gradients (DSSGANs, Transformers, DCL) [54, 86, 130]	◑	●	●	◕	◕	◑	◑	◕	●	◕	●	◕	●
Parameters (Transformers, DSSGANs, DCL) [127, 128, 159]	◑	●	●	●	●	◑	◕	●	◕	●	◕	●	◕
Classification Logits (DSSGANs) [155, 204]	◕	◕	◑	◑	●	○	○	◕	◕	◕	●	◕	●
Task-Specific Head and Tail Parameters (Transformers) [64, 130]	◑	◑	◑	◕	◕	◑	◑	◑	◑	◑	◕	●	◕
Intermediate Data Representations (DCL) [122, 174]	◑	◕	◕	◕	●	◕	◑	◕	◑	◕	●	◕	●
Pretext Logits (DCL) [50, 174]	◕	◕	◑	◕	●	◑	○	●	◕	◑	◕	●	◕

Table 4. Attack types: Data Poisoning (DP), Model Reconstruction (MR), Gradient Leakage (GL), Parameter Inference (PI), Data Inference (DI), Graph Reconstruction (GR), Model Poisoning (MP), Feature Estimation (FE), Adversarial Attacks (AA). Defensive measures: Weighted Steiner Tree (WS), Obfuscation (OB), Privacy-preserving Embedding Sharing (PS), and Secure Representation Sharing (SR). Vulnerability levels of knowledge components (● - critically exposed, ◕ - significant risk, ◑ - attack is possible but harder to exploit, ◔ - minor attack surface, ○ - mostly resistant) and effectiveness of defensive strategies (● - highly effective, ◕ - significantly reduces risk, ◑ - provides partial protection, ◔ - limited effectiveness, ○ - minimal protection).

When classification logits or *task specific head and tail parameters* are exchanged (as in distributed transformers [64, 130]), attackers may exploit *model poisoning* [16] or *feature estimation* [16] techniques. Similarly, *pretext logits* in distributed contrastive learning [50, 174] can be used to perform *feature estimation* [16] or even *adversarial attacks* [50] if intercepted. Moreover, *intermediate data representations* in DCL settings [122, 174] face *graph reconstruction* [155, 204] and *data poisoning* [155] threats when shared among multiple nodes.

To counter these risks, various defensive measures have been proposed. *Differential privacy* [155] is frequently employed to mask sensitive gradients or embedding information, mitigating *gradient leakage* and *parameter inference* [109]. Encrypting gradients [109] further reduces exposure during communication, while *secure parameter sharing* [128] and *privacy preserving embedding sharing* [130] help safeguard classification logits and intermediate representations. Techniques like *Bad Data Detection* [2] use weighted Steiner tree algorithms [87] to detect and isolate malicious updates, minimizing the impact of *model poisoning* [16] on Semi supervised training. Obfuscation [128] and *pipelining with secure aggregation* [92] further protect Transformers' head and tail parameters, while *secure multi-party computation* [122] fortifies representation sharing in DCL against adversarial interception.

Table 4 provides a comprehensive overview of *knowledge components*, their associated *attack vectors* and *defense mechanisms* specifically tailored for Distributed Semi supervised Learning architectures. By integrating these defenses, DSSL can mitigate vulnerabilities inherent in sharing gradients, parameters, and representations, without completely sacrificing model performance.

This section examines knowledge sharing in DSSL, focusing on architectures like DSSGANs, Distributed Transformers, and Contrastive Learning models.

- By combining labelled and unlabelled data, these models enhance generalisation but also introduce risks through shared gradients, embeddings, and attention weights.
- These shared components may encode semantic patterns that can reveal private attributes or reconstruct inputs, especially problematic in settings where anomalies are harder to detect due to limited labels.
- Key threats include data poisoning, gradient leakage, and feature inference. Defenses such as Weighted Steiner Trees, Privacy-Preserving Embedding Sharing, and Secure Representation Sharing are explored as mitigation strategies.

6 Knowledge Sharing in Distributed Deep Reinforcement Learning

Reinforcement Learning has become an essential learning paradigm in AI research. This learning mechanism is crucial for making reliable predictions in more complex and dynamic environments [81]. Since it makes predictions on the dynamic environments, the intelligent agents associated with the learning need to take actions based on the environment in order to maximize the notion of rewarding desired behaviors. The reinforcement learning mechanism is mainly based on making decisions sequentially, and its decisions are dependent on the output of the previous input. *Deep Reinforcement Learning* extended to a distributed setting to make the model operate the way it desired. All the components that need to make a

Knowledge Component	Attacks									Defenses		
	AP	PV	AA	RM	GL	RR	MI	MC	GDP	DP	SA	AT
Policy Weights (DDRL) [36, 117]	◑	◕	●	◑	◕	◕	◕	◑	◕	◕	●	○
Reward Memory (DDRL) [69, 117]	○	◑	◕	●	○	○	◕	◑	○	◑	○	◑
Gradients (DDRL) [74, 113]	◕	◑	◑	○	●	◕	◕	○	●	●	●	◕
State-Action Pairs (DDRL) [69, 74]	◑	●	◕	◕	◑	◑	●	◑	◑	◑	◑	◑
Aggregated Gradients (DDRL) [36, 74]	◑	◑	◕	○	●	●	◕	○	●	●	●	◕
Replay Buffer Data (GORILA, IMPALA) [113, 117]	◕	◑	●	●	◕	◕	◑	◑	◑	◕	◑	◕
Policy Components (IMPALA, SEED RL) [74, 113, 114]	◕	●	●	◕	◑	◕	●	◑	◕	◑	◕	◕
Experience Data (DDRL) [74, 113, 114]	◕	◕	◕	●	◑	◕	●	◕	◑	◑	◑	◕

Table 5. Attack types: Adversarial Perturbations (AP), Policy Value Manipulation (PV), Adversarial Attacks (AA), Reward Manipulation (RM), Gradient Leakage (GL), Recursive Reconstruction (RR), Monitoring and Inference (MI), Memory Corruption (MC), Gradient Disaggregation (GDP). Defensive measures: Differential Privacy (DP), Secure Aggregation (SA), and Adversarial Training (AT). Vulnerability levels of knowledge components (● - critically exposed, ◕ - significant risk, ◑ - attack is possible but harder to exploit, ◔ - minor attack surface, ○ - mostly resistant) and effectiveness of defensive strategies (● - highly effective, ◕ - significantly reduces risk, ◑ - provides partial protection, ◔ - limited effectiveness, ○ - minimal protection).

culminative reward should be shared among the participants alongside the components that are essential for the parallelization of the distributed nodes.

6.1 Distributed Deep Reinforcement Learning Architectures

When general neural networks are combined with a reinforcement learning framework, it is known as *Deep Reinforcement Learning*. This framework efficiently achieves its goals by incorporating techniques such as optimization, function approximation, mapping, and rewards that enhance the performance of the model. This method combines various techniques to make it easier for users to reach their objectives without significant effort. Deep reinforcement learning also utilize algorithms that identify the most efficient path to achieving predefined goals in order to facilitate their learning. In a distributed setting, it is known as *Distributed Deep reinforcement learning (DDRL)*.

DDRL architectures are continuously evolving. The GORILA architecture [117] is a prime example of DDRL, comprising a parameter server, worker, learner, and replay buffer, and relying on the Deep Q-Network (DQN) algorithm [114] to estimate Q-values. Distributed Proximal Policy Optimization (PPO) [69] leverages multiple agents to collect data concurrently for more efficient training, while the Actor-Critic framework [113] mitigates some of GORILA's drawbacks by enabling asynchronous parallel data generation. IMPALA [36] diverges from traditional gradient calculation by transmitting trajectories of experience—comprising active state, current state, reward, and memory to a global model for optimization.

Additionally, approaches like Ape-X [74] and R2D2 [113] implement Recurrent Replay Distributed Reinforcement Learning to reduce variance and accelerate convergence through a gradient prioritization scheme. Finally, SEED RL [35] scales reinforcement learning to larger environments by improving sample efficiency with multiple parallel actors and learners, addressing similar challenges found in IMPALA.

6.2 The Attacks and Defenses on DDRL

Distributed Deep reinforcement learning (DDRL) is vulnerable to attacks targeting key knowledge components such as policy weights [36, 117], reward memory [69, 117], gradients [74, 113], state-action pairs [69, 74], and experience data [74, 113, 114]. These are exploited through adversarial perturbations [36], policy and reward manipulation [117], and replay buffer poisoning [113], leading to skewed learning or model sabotage. Interception of policy components can also compromise decision processes [77, 151], while gradient leakage and model poisoning [113], as well as inference in state action pairs [69], enable subtle degradation of learning performance.

To mitigate these threats, defensive strategies such as differential privacy [51], perturbation [168], local model updates [145], gradient prioritization [36], and secure aggregation [145] are employed. Techniques like buffer sanitization [168] further protect against poisoning by filtering malicious data. Collectively, these methods reinforce DDRL by ensuring how experience data, state action mappings, and policy parameters are shared and updated. Table 5 outlines these components, attacks, and defenses.

This section investigates knowledge sharing in distributed deep reinforcement learning (DDRL) frameworks.

- Vulnerabilities arise from shared elements like policy weights, gradients, and replay buffers, which expose models to risks due to frequent updates and cumulative learning.
- Attacks such as reward manipulation and gradient disaggregation exploit these components to distort training or extract sensitive information.
- Defenses include adversarial training, secure aggregation, differential privacy, and buffer sanitisation, offering layered protection across the reinforcement learning pipeline.

Knowledge Component	Attacks							Defenses		
	DL	PU	BA	MP	PL	KTI	MT	HE	SS	DP
Shared Model Parameters (DTL) [3, 52, 169]	◕	●	◕	●	◑	◕	◔	◕	◑	●
Model Alignment Transfer Components(PPDTL) [53, 138, 191]	◑	◕	◕	◕	●	●	◑	◑	●	◕
Source/Target Domain Data with Model Updates (End-to-End PPDTL) [44, 138]	●	◕	◑	◕	◕	●	◕	◕	◕	●

Table 6. Attack types: Data Leakage (DL), Poisoned Updates (PU), Backdoor Attacks (BA), Model Poisoning (MP), MMD Privacy Leaks (PL), Knowledge Transfer Inference (KTI), Misaligned Transfer (MT). Defensive measures: Homomorphic Encryption (HE), Secret Sharing (SS), and Differential Privacy (DP). Vulnerability levels of knowledge components (● - critically exposed, ◕ - significant risk, ◑ - attack is possible but harder to exploit, ◔ - minor attack surface, ○ - mostly resistant) and effectiveness of defensive strategies (● - highly effective, ◕ - significantly reduces risk, ◑ - provides partial protection, ◔ - limited effectiveness, ○ - minimal protection).

7 Knowledge Sharing in Distributed Transfer Learning

Transfer Learning is a machine learning technique where a model developed for a specific task is reused to improve performance on a related task [125]. It can be applied to various learning tasks, including supervised [116], unsupervised [207], semi-supervised [23], and reinforcement learning [213].

Distributed transfer learning (DTL) combines transfer learning with distributed learning principles [170], particularly for edge resource utilization in knowledge sharing and task offloading [195]. Xu et al. [178] present a DTL model based on multisource heterogeneous transfer learning, incorporating Domain Specific Embedding (DSE) and Global Shared Embedding (GSE). DSE is unique to each domain, while GSE captures features across all domains.

Hashemian et al. [61] contribute with two architectures: PPDUSTR (Privacy Preserving Unsupervised Transfer Learning) and PPDESTR (Privacy Preserving Distributed Semi-supervised Transfer Learning). PPDUSTR trains a Privacy Preserving Support Vector Machine (PPSVM) on a server, ensuring client data privacy, while PPDESTR addresses privacy concerns in semi-supervised learning through techniques like pseudo-labeling. Additionally, Mignone et al. [112] introduce DISHTRA (distributed heterogeneous transfer learning) for link prediction tasks in Positive Unlabeled (PU) learning, using the MapReduce model with Apache Spark to handle large datasets across multiple nodes.

7.1 Knowledge Components in Distributed Transfer Learning

In DTL, *knowledge components* being transferred often include *shared model parameters*, *transfer components* (e.g., embedding layers or domain adaptation modules), and *domain embeddings* (DSE and GSE). By leveraging pre trained models or transfer modules, DTL can effectively reuse knowledge to boost performance on related tasks in different domains. PPDUSTR and PPDESTR introduce privacy-preserving protocols that handle *model alignment* and *transfer component analysis (TCA)* without exposing raw data [61]. DISHTRA focuses on *large scale PU learning* by distributing *domain embeddings* and *mapping functions* across multiple nodes [112].

These shared components allow nodes to coordinate tasks such as classification, clustering, or link prediction while minimizing communication overhead. *Heterogeneous* or *multi source* transfer architectures [178] rely on unique and global embeddings (DSE/GSE) to capture domain invariant features, thereby enhancing adaptability and preserving performance in distributed environments.

7.2 Attacks and Defenses in Distributed Transfer Learning

Despite the benefits of knowledge reuse, DTL is susceptible to *data leakage*, *poisoned updates*, *backdoor attacks*, and *model poisoning* [3, 52, 169]. For instance, an adversary intercepting *shared model parameters* or *transfer components* may reconstruct sensitive data or introduce malicious perturbations. Further, *model misalignment* and *inference* attacks arise from insecure domain adaptation mechanisms (e.g., MMD-based approaches) [53].

As mitigation, *Privacy Preserving Distributed Transfer Learning (PPDTL)* protocols [98], *homomorphic encryption (HE)* [138], and *secret sharing (SS)* [44] are widely adopted. These approaches encrypt parameters or domain embeddings before exchange, ensuring secure computation under partial or fully homomorphic conditions. Additionally, DP can be applied to obfuscate sensitive gradients or embeddings, reducing the likelihood of inference attacks. For MMD based methods, *Secure MMD (SMMD)* [191] and *HE for MMD loss* [138] defend against *MMD privacy leaks* by preventing direct access to raw distributions. Table 6 summarizes the main *knowledge components*, potential *attacks*, and *defense techniques* across DTL architectures, highlighting how secure transfer protocols ranging from partially homomorphic encryption (PHE) to advanced DP schemes help maintain data security and privacy while exploiting the advantages of reuse in distributed learning setups.

This section explores knowledge sharing in DTL architectures such as PPDUSTR, PPDESTR, and DISHTRA.
- The architectures reuse components like embeddings and transfer modules across tasks and nodes.
- Shared embeddings and transfer modules could become common targets for attacks, such as backdoors and misalignment-based inference.
- To mitigate, defenses such as HE, secret sharing, and secure MMD techniques have been explored in the literature.

8 Distributed Learning in Large Language Models

The advancements in large language models (LLMs) have revolutionized a wide range of domains, from natural language processing (NLP) to conversational AI. Models such as GPT and BERT leverage massive datasets and high-performance computing infrastructures to achieve state-of-the-art results in understanding and generating human language. However, their centralized training paradigm raises concerns about privacy, data governance, and scalability, especially when sensitive or proprietary data is distributed across multiple entities. To address these challenges, the concept of *decentralized LLMs* has emerged, where model training and inference are performed collaboratively between distributed nodes without requiring direct data sharing.

This paradigm enhances privacy preservation and compliance with data sovereignty regulations, while also enabling participation from resource-constrained stakeholders. Nonetheless, decentralized LLMs introduce new complexities in synchronization, communication overhead, and ensuring consistency across heterogeneous environments. Balancing these factors is critical to realizing secure, efficient, and inclusive language model ecosystems.

8.1 Distributed LLM Architectures

8.1.1 Partial Fine Tuning. Partial fine tuning targets specific components of a pretrained LLM, such as attention weights or output embeddings, while leaving foundational layers intact. This method reduces computational overhead and preserves data privacy, as only updated parameters are shared [208]. SplitLoRA [100] exemplifies this approach by splitting training between clients and a server, yielding faster and more accurate fine tuning. As an open source benchmark for split learning (SL) in LLMs, SplitLoRA effectively demonstrates how partial fine tuning enhances model performance without extensive resource demands.

8.1.2 Collaborative Agreement on What to Share. Collaboration mechanisms in distributed LLMs focus on selectively sharing gradients, weights, or feature representations to respect data sovereignty and satisfy privacy regulations [180]. This approach promotes trust among participants while maintaining robust collective learning [79]. Frameworks like CLLM4Rec [212] and CoLLM [201] illustrate how user and item embeddings, textual features, and interaction data are transformed into token sequences or treated as a separate modality, respectively. By clearly defining the shared components, these methods seamlessly integrate collaborative information into LLMs for improved recommendation capabilities.

8.1.3 Periodic Update Sharing. Periodic update sharing in LLMs involves exchanging updates at predefined intervals to address evolving domain specific or linguistic needs. NewTerm [28] demonstrates this by releasing benchmarks for newly coined terms, allowing annual performance evaluations of LLMs on emerging vocabulary. Similarly, the Self Information Update (SIU) task [187] periodically integrates fresh textual data, instruction response pairs, and distillation strategies into a distributed LLM framework. By bundling these updates, SIU ensures consistent performance improvements and mitigates outdated content in long running models.

8.1.4 RAG with Distributed Learning. Retrieval Augmented Generation (RAG) integrates an external retrieval mechanism into LLM inference, enabling the model to access relevant knowledge from a distributed corpus during decoding. In distributed learning environments, RAG architectures benefit from horizontally partitioned knowledge sources across multiple nodes, where each participant contributes domain specific documents or embeddings. This distributed RAG paradigm enhances factual grounding and supports low latency responses by parallelizing retrieval and generation. Recent systems leverage secure retrieval protocols to mitigate information leakage, ensuring that query embeddings and retrieved content remain private during collaboration [103].

8.1.5 Federated Fine Tuning. Federated fine tuning relies on decentralized datasets to tailor a pre trained LLM to specific tasks. FlexLoRA [7] leverages Low Rank Adaptation (LoRA) configurations and dynamically adjusts local model ranks to accommodate client resource constraints, aggregating them into a global model via Singular Value Decomposition (SVD). FLoRA [165] further refines federated fine tuning by adopting a noise free aggregation mechanism for heterogeneous LoRA updates. These frameworks enable distributed clients to share only the essential weight matrices and model insights, preserving efficiency and adaptability in large scale LLM deployments.

8.2 Knowledge Sharing in Distributed LLMs: Attacks and Defenses

Distributed LLMs rely on the exchange of specific knowledge components to enable the development of collaborative models while maintaining privacy and efficiency. As discussed earlier, methods such as partial fine-tuning and frameworks such as SplitLoRA [100] and FlexLoRA [165] optimize resource usage by sharing only updated parameters, attention weights, or LoRA weight matrices.

Collaborative agreements similarly govern the selective sharing of gradients, user-item embeddings, textual features, and historical interaction data to improve model performance while adhering to privacy regulations. However, these shared components can be exploited by sophisticated attacks, posing risks to both the integrity of the model and the privacy of the contributor. Model inversion attacks, for instance, can exploit shared embeddings or gradients to reconstruct sensitive local datasets.

For example, user-item embeddings shared in frameworks like CLLM4Rec [212] and CoLLM [201], or newly sourced textual information from the Self-Information Update (SIU) [187] task, could be reverse engineered to extract private data about users or contributors. Furthermore, data poisoning attacks can target periodic update sharing mechanisms, such as those in NewTerm [28] , by injecting malicious data that propagates through the system, corrupting collaborative models and shared benchmarks.

To counter such threats, privacy-preserving techniques must be tightly integrated with model-sharing mechanisms. For example, applying DP to shared knowledge components, such as LoRA weights or collaborative embeddings, can obscure sensitive information through controlled noise. In frameworks such as FlexLoRA and FLoRA [165], this involves adding DP to LoRA ranks or the SVD-based aggregation [60], enabling secure sharing of task-specific insights without exposing private data.

9 Limitations of Current Knowledge Sharing Schemes and Future Directions

Across distributed learning paradigms DSL, DUSL, DSSL, DDRL, DTL, and distributed LLM, we observe a consistent challenge: component level knowledge sharing significantly enhances learning utility, but inherently expands the system's attack surface. No single defense mechanism—encryption, differential privacy (DP), secure multiparty computation (SMPC), trusted execution environments (TEEs), or obfuscation—can offer comprehensive protection. This underscores the growing necessity for hybrid approaches that intelligently combine multiple techniques.

9.1 Preserving Privacy While Maximizing Utility

Defensive mechanisms used in isolation impose major trade-offs: encryption introduces latency, DP compromises accuracy, SMPC inflates communication overhead, and obfuscation reduces model interpretability. To address these constraints, hybrid approaches have emerged that layer complementary protections on the most sensitive knowledge components.

Encryption-centric hybrids, such as SMPC + (P)HE Participants first secret-share model updates, then encapsulate these shares using partially homomorphic encryption. Aggregation is performed directly in the encrypted domain, minimizing plaintext exposure while keeping communication costs manageable. This method is especially suitable for DDRL replay buffer statistics or DTL embeddings where exact aggregation is essential.

Noise-centric hybrids such as TEE + DP/HE Model updates are decrypted only within the enclave memory. Inside the TEE, lightweight DP or fine-grained HE operations are applied before releasing the results. In SplitLoRA-like settings, LoRA matrices are shielded within the enclave and further perturbed with calibrated DP noise, ensuring privacy even in the event of host compromise.

9.1.1 Adaptive Orchestration. Despite progress, production environments lack *adaptive privacy orchestrators*—systems capable of dynamically adjusting noise levels or encryption depth based on adversarial activity, system load, or drift in data distribution; ranking knowledge components by risk and selectively applying heavier protections to high-leakage targets and coordinating privacy policies across device hierarchies without disrupting ongoing training. Such orchestration would require real-time feedback through privacy loss estimators, throughput monitors, and threat signals to adjust parameters, rotate keys, or reconfigure learning splits on demand.

9.1.2 Cost Modeling and Benchmarking. A comprehensive cost model for hybrid defenses remains absent, prompting the need for a unified framework to evaluate their effectiveness. We propose a joint metric framework where utility is defined as a function of accuracy and convergence, privacy is measured in terms of differential privacy parameters and ciphertext entropy, and overhead is evaluated through latency, accuracy, and bit operations. Standardizing these metrics across various learning paradigms is essential to reveal the inherent trade-offs between privacy, utility, and efficiency. This standardization can guide the design of optimally balanced hybrid architectures that align with both business objectives and regulatory requirements.

9.2 Towards Unified, Modular Privacy Frameworks

As detailed throughout Sections 3–8, current systems lack a modular privacy stack that supports plug-and-play hybrid defenses for components such as gradients (DSL), logits (DUSL), LoRA matrices (LLMs), or trajectories (DDRL). A consistent API offering modular primitives for encryption, DP, SMPC, and TEE offloading would:

- enhance reproducibility by decoupling defense mechanisms from model training logic;
- support compliance by embedding auditability and privacy accounting by design;
- offer a sandbox for performance benchmarking across heterogeneous systems.

9.3 Open Research Questions

Our systematic review identifies several open research questions.

RQ1: How can privacy orchestrators be designed to dynamically co-optimize differential privacy budgets, cryptographic parameters, and secure multiparty computation (SMPC) participation in response to real-time workload changes, latency constraints, and varying privacy risk levels in distributed AI systems?

RQ2: Which benchmark configurations—across datasets, model types, and adversarial setups—most effectively quantify privacy–utility trade-offs for modular defenses against attacks such as model inversion, gradient leakage, and replay poisoning in distributed learning systems?

RQ3: What interface requirements and abstraction boundaries enable seamless integration and replacement of evolving privacy-preserving mechanisms—such as DP, HE, or SMPC—without retraining models or violating emerging AI regulations like GDPR or the EU AI Act?

Addressing these challenges will be essential for achieving scalable, adaptive, and regulation-compliant privacy frameworks in distributed AI.

10 Conclusion

This paper has offered a comprehensive analysis of the diverse types of *knowledge* that can be exchanged during collaborative AI/ML operations among distributed entities. A key insight is that knowledge sharing in distributed learning is context-dependent, shaped by architecture, learning goals, and collaboration. We observed that no single defense mechanism is universally effective. Instead, each distributed learning paradigm benefits from tailored strategies. For example, techniques such as differential privacy and secure aggregation prove consistently valuable across supervised and semi-supervised settings, while obfuscation, adversarial training, and privacy preserving embedding sharing become more relevant in unsupervised, reinforcement, and LLM based architectures. The paper highlights the need for component-level privacy integration and introduces a multi-dimensional evaluation framework, laying the foundation for unified, adaptable privacy preserving solutions in increasingly sensitive distributed learning environments.

Acknowledgement

We thank the reviewers for their constructive feedback, which helped improve the quality of this work. This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

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] Mohammad Hasan Ansari, Vahid Tabataba Vakili, Behnam Bahrak, and Parmiss Tavassoli. 2018. Graph theoretical defense mechanisms against false data injection attacks in smart grids. *Journal of Modern Power Systems and Clean Energy* 6, 5 (2018), 860–871.

[3] Mohammad S Ansari, Saeed H Alsamhi, Yuansong Qiao, Yuhang Ye, and Brian Lee. 2020. Security of distributed intelligence in edge computing: Threats and countermeasures. *The Cloud-to-Thing Continuum: Opportunities and Challenges in Cloud, Fog and Edge Computing* (2020), 95–122.

[4] Yoshinori Aono, Takuya Hayashi, Lihua Wang, Shiho Moriai, et al. 2017. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Transactions on Information Forensics and Security* 13, 5 (2017), 1333–1345.

[5] Rezak Aziz, Soumya Banerjee, Samia Bouzefrane, and Thinh Le Vinh. 2023. Exploring homomorphic encryption and differential privacy techniques towards secure federated learning paradigm. *Future internet* 15, 9 (2023), 310.

[6] 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.

[7] Jiamu Bai, Daoyuan Chen, Bingchen Qian, Liuyi Yao, and Yaliang Li. 2024. Federated fine-tuning of large language models under heterogeneous language tasks and client resources. *arXiv e-prints* (2024), arXiv–2402.

[8] Li Bai, Haibo Hu, Qingqing Ye, Haoyang Li, Leixia Wang, and Jianliang Xu. 2024. Membership Inference Attacks and Defenses in Federated Learning: A Survey. *Comput. Surveys* 57, 4 (2024), 1–35.

[9] Maria Florina Balcan, Avrim Blum, Shai Fine, and Yishay Mansour. 2012. Distributed Learning, Communication Complexity and Privacy. In *Proceedings of the 25th Annual Conference on Learning Theory*, Vol. 23. PMLR, 26.1–26.22.

[10] Gilad Baruch, Moran Baruch, and Yoav Goldberg. 2019. A little is enough: Circumventing defenses for distributed learning. *Advances in Neural Information Processing Systems* 32 (2019).

[11] Vahid Behzadan and Arslan Munir. 2017. Vulnerability of deep reinforcement learning to policy induction attacks. In *Machine Learning and Data Mining in Pattern Recognition: 13th International Conference, MLDM 2017, New York, NY, USA, July 15-20, 2017, Proceedings 13*. Springer, 262–275.

[12] Battista Biggio, Blaine Nelson, and Pavel Laskov. 2012. Poisoning attacks against support vector machines. *arXiv preprint arXiv:1206.6389* (2012).

[13] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2017. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (2017), 1175–1191.

[14] Eitan Borgnia, Valeriia Cherepanova, Liam Fowl, Amin Ghiasi, Jonas Geiping, Micah Goldblum, Tom Goldstein, and Arjun Gupta. 2021. Strong data augmentation sanitizes poisoning and backdoor attacks without an accuracy tradeoff. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 3855–3859.

[15] Wadii Boulila, Mokhtar Sellami, Maha Driss, Mohammed Al-Sarem, Mahmood Safaei, and Fuad A Ghaleb. 2021. RS-DCNN: A novel distributed convolutional-neural-networks based-approach for big remote-sensing image classification. *Computers and Electronics in Agriculture* 182 (2021), 106014.

[16] Nicholas Carlini. 2021. Poisoning the Unlabeled Dataset of Semi-Supervised Learning. In *Proceedings of the 30th USENIX Security Symposium (USENIX Security 21)* (2021), 1577–1592.

[17] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX security symposium (USENIX security 19)*. 267–284.

[18] Juan Cervino, Md Asadullah Turja, Hesham Mostafa, Nageen Himayat, and Alejandro Ribeiro. 2024. Distributed training of large graph neural networks with variable communication rates. *arXiv preprint arXiv:2406.17611* (2024).

[19] Edward Y Chang, Shih-Wei Liao, Chun-Ting Liu, Wei-Chen Lin, Pin-Wei Liao, Wei-Kang Fu, Chung-Huan Mei, and Emily J Chang. 2018. DeepLinQ: distributed multi-layer ledgers for privacy-preserving data sharing. In *Proceedings of 2018 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)* (2018), 173–178.

[20] Nandish Chattopadhyay, Amira Guesmi, Muhammad Abdullah Hanif, Bassem Ouni, and Muhammad Shafique. 2024. Defending against Adversarial Patches using Dimensionality Reduction. In *Proceedings of the 61st ACM/IEEE Design Automation Conference*. 1–6.

[21] Mingzhe Chen, Deniz Gündüz, Kaibin Huang, Walid Saad, Mehdi Bennis, Aneta Vulgarakis Feljan, and H Vincent Poor. 2021. Distributed learning in wireless networks: Recent progress and future challenges. *IEEE Journal on Selected Areas in Communications* 39, 12 (2021), 3579–3605.

[22] Peng Chen, Xin Du, Zhihui Lu, and Hongfeng Chai. 2023. Universal adversarial backdoor attacks to fool vertical federated learning in cloud-edge collaboration. *arXiv preprint arXiv:2304.11432* (2023).

[23] Veronika Cheplygina, Marleen De Bruijne, and Josien PW Pluim. 2019. Not-so-supervised: a survey of semi-supervised, multi-instance, and transfer learning in medical image analysis. *Medical image analysis* 54 (2019), 280–296.

[24] Konstantina Christakopoulou and Arindam Banerjee. 2019. Adversarial attacks on an oblivious recommender. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 322–330.

[25] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. 2018. Generative adversarial networks: An overview. *IEEE Signal Processing Magazine* 35, 1 (2018), 53–65.

[26] Hanjun Dai, Hui Li, Tian Tian, Xin Huang, Lin Wang, Jun Zhu, and Le Song. 2018. Adversarial attack on graph structured data. In *International conference on machine learning*. PMLR, 1115–1124.

[27] Patrick Dendorfer, Sven Elflein, and Laura Leal-Taixé. 2021. MG-GAN: A multi-generator model preventing out-of-distribution samples in pedestrian trajectory prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), 13158–13167.

[28] Hexuan Deng, Wenxiang Jiao, Xuebo Liu, Min Zhang, and Zhaopeng Tu. 2024. NewTerm: Benchmarking real-time new terms for large language models with annual updates. *arXiv preprint arXiv:2410.20814* (2024).

[29] Enmao Diao, Jie Ding, and Vahid Tarokh. 2020. Drasic: Distributed recurrent autoencoder for scalable image compression. In *Proceedings of the 2020 Data Compression Conference (DCC)* (2020), 3–12.

[30] Yi Dong, Yang Chen, Xingyu Zhao, and Xiaowei Huang. 2022. Short-term Load Forecasting with Distributed Long Short-Term Memory. *arXiv preprint arXiv:2208.01147* (2022).

[31] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[32] Thang Le Duc, Rafael García Leiva, Paolo Casari, and Per-Olov Östberg. 2019. Machine learning methods for reliable resource provisioning in edge-cloud computing: A survey. *ACM Computing Surveys (CSUR)* 52, 5 (2019), 1–39.

[33] 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.

[34] Ayman S. El-Baz and Jasjit S. Suri. 2021. Chapter 8 - Machine learning methods for autism spectrum disorder classification. In *Proceedings of the Neural Engineering Techniques for Autism Spectrum Disorder* (2021), 151–163. https://doi.org/10.1016/B978-0-12-822822-7.00008-9

[35] Lasse Espeholt, Raphaël Marinier, Piotr Stanczyk, Ke Wang, and Marcin Michalski. 2019. SEED RL: Scalable and efficient deep-RL with accelerated central inference. *arXiv preprint arXiv:1910.06591* (2019).

[36] Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Vlad Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, et al. 2018. Impala: Scalable distributed deep-RL with importance weighted actor-learner architectures. In *Proceedings of the International Conference on Machine Learning* (2018), 1407–1416.

[37] Minghong Fang, Neil Zhenqiang Gong, and Jia Liu. 2020. Influence function based data poisoning attacks to top-n recommender systems. In *Proceedings of The Web Conference 2020*. 3019–3025.

[38] Yunhao Feng, Yanming Guo, Yinjian Hou, Yulun Wu, Mingrui Lao, Tianyuan Yu, and Gang Liu. 2025. A survey of security threats in federated learning. *Complex & Intelligent Systems* 11, 2 (2025), 1–26.

[39] Carlos Poncinelli Filho, Elias Marques, Victor Chang, Leonardo dos Santos, Flavia Bernardini, Paulo F. Pires, Luiz Ochi, and Flavia C. Delicato. 2022. A Systematic Literature Review on Distributed Machine Learning in Edge Computing. *Sensors* 22, 7 (2022), 1–36. https://www.mdpi.com/1424-8220/22/7/2665

[40] David Froelicher, Juan R Troncoso-Pastoriza, Apostolos Pyrgelis, Sinem Sav, Joao Sa Sousa, Jean-Philippe Bossuat, and Jean-Pierre Hubaux. 2021. Scalable privacy-preserving distributed learning. In *Proceedings of the Privacy Enhancing Technologies* 2021, 2 (2021), 323–347.

[41] Chong Fu, Xuhong Zhang, Shouling Ji, Jinyin Chen, Jingzheng Wu, Shanqing Guo, Jun Zhou, Alex X Liu, and Ting Wang. 2022. Label inference attacks against vertical federated learning. In *31st USENIX security symposium (USENIX Security 22)*. 1397–1414.

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

[43] Shripad Gade and Nitin H Vaidya. 2018. Privacy-preserving distributed learning via obfuscated stochastic gradients. In *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 184–191.

[44] Dashan Gao, Yang Liu, Anbu Huang, Ce Ju, Han Yu, and Qiang Yang. 2019. Privacy-preserving heterogeneous federated transfer learning. In *2019 IEEE international conference on big data (Big Data)*. IEEE, 2552–2559.

[45] Matt W Gardner and SR Dorling. 1998. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric Environment* 32, 14-15 (1998), 2627–2636.

[46] Luis Garrote, Lúcia Martins, Urbano J Nunes, and Martin Zachariasen. 2019. Weighted euclidean steiner trees for disaster-aware network design. In *2019 15th International Conference on the Design of Reliable Communication Networks (DRCN)*. IEEE, 138–145.

[47] 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.

[48] Marcel Geppert, Viktor Larsson, Johannes L Schönberger, and Marc Pollefeys. 2022. Privacy preserving partial localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 17337–17347.

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

[50] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. 2018. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728* (2018).

[51] Parham Gohari, Bo Chen, Bo Wu, Matthew Hale, and Ufuk Topcu. 2021. Privacy-preserving kickstarting deep reinforcement learning with privacy-aware learners. *arXiv preprint arXiv:2102.09599* (2021).

[52] Eduard Gorbunov, Alexander Borzunov, Michael Diskin, and Max Ryabinin. 2022. Secure distributed training at scale. In *International Conference on Machine Learning*. PMLR, 7679–7739.

[53] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. A kernel two-sample test. *The Journal of Machine Learning Research* 13, 1 (2012), 723–773.

[54] Hongtao Guan, Xingkong Ma, and Siqi Shen. 2020. DOS-GAN: A Distributed Over-Sampling Method Based on Generative Adversarial Networks for Distributed Class-Imbalance Learning. In *Proceedings of the International Conference on Algorithms and Architectures for Parallel Processing* (2020), 609–622.

[55] Rachid Guerraoui, Sébastien Rouault, et al. 2018. The hidden vulnerability of distributed learning in byzantium. In *Proceedings of the International Conference on Machine Learning* (2018), 3521–3530.

[56] Shangwei Guo, Xu Zhang, Fei Yang, Tianwei Zhang, Yan Gan, Tao Xiang, and Yang Liu. 2021. Robust and privacy-preserving collaborative learning: A comprehensive survey. *arXiv preprint arXiv:2112.10183* (2021).

[57] Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)* 2 (2006), 1735–1742.

[58] Kai Han, Yunhe Wang, Chao Zhang, Chao Li, and Chao Xu. 2018. Autoencoder inspired unsupervised feature selection. In *Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2018), 2941–2945.

[59] Corentin Hardy, Erwan Le Merrer, and Bruno Sericola. 2019. MD-GAN: Multi-discriminator generative adversarial networks for distributed datasets. In *Proceedings of the 2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS)* (2019), 866–877.

[60] Anne Hartebrodt, Richard Röttger, and David B Blumenthal. 2024. Federated singular value decomposition for high-dimensional data. *Data Mining and Knowledge Discovery* 38, 3 (2024), 938–975.

[61] Mina Hashemian, Farbod Razzazi, Houman Zarrabi, and Mohammad Shahram Moin. 2021. Semi-supervised and unsupervised privacy-preserving distributed transfer learning approach in HAR systems. *Wireless Personal Communications* 117 (2021), 637–654.

[62] Muneeb Ul Hassan, Mubashir Husain Rehmani, and Jinjun Chen. 2019. Differential privacy techniques for cyber physical systems: a survey. *IEEE Communications Surveys & Tutorials* 22, 1 (2019), 746–789.

[63] Chaoyang He, Murali Annavaram, and Salman Avestimehr. 2020. Group knowledge transfer: Federated learning of large cnns at the edge. *Advances in neural information processing systems* 33 (2020), 14068–14080.

[64] Chaoyang He, Shen Li, Mahdi Soltanolkotabi, and Salman Avestimehr. 2021. Pipetransformer: Automated elastic pipelining for distributed training of transformers. *arXiv preprint arXiv:2102.03161* (2021).

[65] Xinlei He, Jinyuan Jia, Michael Backes, Neil Zhenqiang Gong, and Yang Zhang. 2021. Stealing links from graph neural networks. In *30th USENIX security symposium (USENIX security 21)*. 2669–2686.

[66] Xuanli He, Lingjuan Lyu, Qiongkai Xu, and Lichao Sun. 2021. Model extraction and adversarial transferability, your BERT is vulnerable! *arXiv preprint arXiv:2103.10013* (2021).

[67] Xinlei He and Yang Zhang. 2021. Quantifying and mitigating privacy risks of contrastive learning. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. 845–863.

[68] Zhi He, Han Liu, Yiwen Wang, and Jie Hu. 2017. Generative adversarial networks-based semi-supervised learning for hyperspectral image classification. *Remote Sensing* 9, 10 (2017), 1042.

[69] Nicolas Heess, Dhruva TB, Srinivasan Sriram, Jay Lemmon, Josh Merel, Greg Wayne, Yuval Tassa, Tom Erez, Ziyu Wang, SM Eslami, et al. 2017. Emergence of locomotion behaviours in rich environments. *arXiv preprint arXiv:1707.02286* (2017).

[70] Vishakh Hegde and Sheema Usmani. 2016. Parallel and distributed deep learning. *ICME, Stanford University* 31 (2016), 1–8.

[71] Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. 2017. Deep models under the GAN: information leakage from collaborative deep learning. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*. 603–618.

[72] Quan Hoang, Tu Dinh Nguyen, Trung Le, and Dinh Phung. 2018. MGAN: Training generative adversarial nets with multiple generators. In *Proceedings of the International Conference on Learning Representations* 30 (2018).

[73] Hyeong Gwon Hong, Yooshin Cho, Hanbyel Cho, Jaesung Ahn, and Junmo Kim. 2024. Foreseeing reconstruction quality of gradient inversion: An optimization perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 12473–12481.

[74] Dan Horgan, John Quan, David Budden, Gabriel Barth-Maron, Matteo Hessel, Hado Van Hasselt, and David Silver. 2018. Distributed prioritized experience replay. *arXiv preprint arXiv:1803.00933* (2018).

[75] Shuyan Hu, Xiaojing Chen, Wei Ni, Ekram Hossain, and Xin Wang. 2021. Distributed machine learning for wireless communication networks: Techniques, architectures, and applications. *IEEE Communications Surveys & Tutorials* 23, 3 (2021), 1458–1493.

[76] Chaolin Huang, Yeming Wang, Xingwang Li, Lili Ren, Jianping Zhao, Yi Hu, Li Zhang, Guohui Fan, Jiuyang Xu, Xiaoying Gu, et al. 2020. Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China. *The lancet* 395, 10223 (2020), 497–506.

[77] Inaam Ilahi, Muhammad Usama, Junaid Qadir, Muhammad Umar Janjua, Ala Al-Fuqaha, Dinh Thai Hoang, and Dusit Niyato. 2021. Challenges and countermeasures for adversarial attacks on deep reinforcement learning. *IEEE Transactions on Artificial Intelligence* 3, 2 (2021), 90–109.

[78] 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.

[79] David Jin, Niclas Kannengießer, Sascha Rank, and Ali Sunyaev. 2024. Collaborative Distributed Machine Learning. *Comput. Surveys* (2024).

[80] James Jordon, Jinsung Yoon, and Mihaela Van Der Schaar. 2018. PATE-GAN: Generating synthetic data with differential privacy guarantees. In *Proceedings of the International Conference on Learning Representations* (2018).

[81] Steven Kapturowski, Georg Ostrovski, John Quan, Remi Munos, and Will Dabney. 2018. Recurrent experience replay in distributed reinforcement learning. (2018).

[82] Fumiyuki Kato, Yang Cao, and Masatoshi Yoshikawa. 2022. Olive: Oblivious federated learning on trusted execution environment against the risk of sparsification. *arXiv preprint arXiv:2202.07165* (2022).

[83] Raouf Kerkouche, Gergely Ács, and Mario Fritz. 2023. Client-specific property inference against secure aggregation in federated learning. In *Proceedings of the 22nd Workshop on Privacy in the Electronic Society*. 45–60.

[84] Ivan Kholod, Andrey Rukavitsyn, Alexey Paznikov, and Sergei Gorlatch. 2021. Parallelization of the self-organized maps algorithm for federated learning on distributed sources. *The Journal of Supercomputing* 77, 6 (2021), 6197–6213.

[85] Minhoe Kim, Souhwan Jung, and Minho Park. 2015. A distributed self-organizing map for DoS attack detection. In *Proceedings of the 2015 Seventh International Conference on Ubiquitous and Future Networks* (2015), 19–22.

[86] Margarita Kirienko, Martina Sollini, Gaia Ninatti, Daniele Loiacono, Edoardo Giacomello, Noemi Gozzi, Francesco Amigoni, Luca Mainardi, Pier Luca Lanzi, and Arturo Chiti. 2021. Distributed learning: a reliable privacy-preserving strategy to change multicenter collaborations using AI. *European Journal of Nuclear Medicine and Molecular Imaging* 48, 12 (2021), 3791–3804.

[87] Philip Klein and RJJA Ravi. 1995. A nearly best-possible approximation algorithm for node-weighted Steiner trees. *Journal of Algorithms* 19, 1 (1995), 104–115.

[88] Teuvo Kohonen and Timo Honkela. 2007. Kohonen network. *Scholarpedia* 2, 1 (2007), 1568.

[89] Abdulkadir Korkmaz and Praveen Rao. 2025. A Selective Homomorphic Encryption Approach for Faster Privacy-Preserving Federated Learning. *arXiv*

preprint arXiv:2501.12911 (2025).

[90] Kalpesh Krishna, Gaurav Singh Tomar, Ankur P. Parikh, Nicolas Papernot, and Mohit Iyyer. 2019. Thieves on Sesame Street! Model Extraction of BERT-based APIs. *CoRR* abs/1910.12366 (2019). arXiv:1910.12366 http://arxiv.org/abs/1910.12366

[91] Abhishek Kumar, Prasanna Sattigeri, and Tom Fletcher. 2017. Semi-supervised learning with GANs: Manifold invariance with improved inference. *Advances in Neural Information Processing Systems* 30 (2017).

[92] Banafsheh Saber Latibari, Najmeh Nazari, Muhtasim Alam Chowdhury, Kevin Immanuel Gubbi, Chongzhou Fang, Sujan Ghimire, Elahe Hosseini, Hossein Sayadi, Houman Homayoun, Soheil Salehi, et al. 2024. Transformers: A Security Perspective. *IEEE Access* (2024).

[93] F Thomson Leighton. 2014. *Introduction to Parallel Algorithms and Architectures: Arrays· Trees· Hypercubes*. Elsevier.

[94] Bo Li, Yining Wang, Aarti Singh, and Yevgeniy Vorobeychik. 2016. Data poisoning attacks on factorization-based collaborative filtering. *Advances in neural information processing systems* 29 (2016).

[95] Li Li, Alfredo Bayuelo, Leonardo Bobadilla, Tauhidul Alam, and Dylan A Shell. 2019. Coordinated multi-robot planning while preserving individual privacy. In *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2188–2194.

[96] Yuqing Li, Nan Yan, Jing Chen, Xiong Wang, Jianan Hong, Kun He, Wei Wang, and Bo Li. 2025. FedPHE: A Secure and Efficient Federated Learning via Packed Homomorphic Encryption. *IEEE Transactions on Dependable and Secure Computing* (2025).

[97] Zhichao Li, Li Tian, Qingchao Jiang, and Xuefeng Yan. 2021. Distributed-ensemble stacked autoencoder model for non-linear process monitoring. *Information Sciences* 542 (2021), 302–316.

[98] Zhi Li, Hao Wang, Guangquan Xu, Alireza Jolfaei, Xi Zheng, Chunhua Su, and Wenying Zhang. 2022. Privacy-preserving distributed transfer learning and its application in intelligent transportation. *IEEE Transactions on Intelligent Transportation Systems* (2022).

[99] 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).

[100] Zheng Lin, Xuanjie Hu, Yuxin Zhang, Zhe Chen, Zihan Fang, Xianhao Chen, Ang Li, Praneeth Vepakomma, and Yue Gao. 2024. Splitlora: A split parameter-efficient fine-tuning framework for large language models. *arXiv preprint arXiv:2407.00952* (2024).

[101] Ji Liu, Jizhou Huang, Yang Zhou, Xuhong Li, Shilei Ji, Haoyi Xiong, and Dejing Dou. 2022. From distributed machine learning to federated learning: A survey. *Knowledge and Information Systems* 64, 4 (2022), 885–917.

[102] Junlin Liu and Xinchen Lyu. 2022. Clustering Label Inference Attack against Practical Split Learning. *arXiv preprint arXiv:2203.05222* (2022).

[103] Shige Liu, Zhifang Zeng, Li Chen, Adil Ainihaer, Arun Ramasami, Songting Chen, Yu Xu, Mingxi Wu, and Jianguo Wang. 2025. TigerVector: Supporting Vector Search in Graph Databases for Advanced RAGs. *arXiv preprint arXiv:2501.11216* (2025).

[104] Jia Lu, Ryan Tsoi, Nan Luo, Yuanchi Ha, Shangying Wang, Minjun Kwak, Yasa Baig, Nicole Moiseyev, Shari Tian, Alison Zhang, et al. 2022. Distributed information encoding and decoding using self-organized spatial patterns. *Patterns* 3, 10 (2022), 100590.

[105] Yecheng Lyu, Lin Bai, and Xinming Huang. 2019. Road segmentation using CNN and distributed LSTM. In *Proceedings of the 2019 IEEE International Symposium on Circuits and Systems (ISCAS)* (2019), 1–5.

[106] Chuan Ma, Jun Li, Kang Wei, Bo Liu, Ming Ding, Long Yuan, Zhu Han, and H Vincent Poor. 2023. Trusted ai in multiagent systems: An overview of privacy and security for distributed learning. *Proc. IEEE* 111, 9 (2023), 1097–1132.

[107] A Madry, A Makelov, L Schmidt, D Tsipras, and A Vladu. 2018. Towards deep learning models resistant to adversarial attacks. 6th Int. In *Conf. Learn. Represent. ICLR*.

[108] Tarek Makkouk, Dong Jae Kim, and Tse-Hsun Peter Chen. 2022. An empirical study on performance bugs in deep learning frameworks. In *2022 ieee international conference on software maintenance and evolution (icsme)*. IEEE, 35–46.

[109] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Proceedigns of the Artificial Intelligence and Statistics* (2017), 1273–1282.

[110] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. 2019. Exploiting unintended feature leakage in collaborative learning. In *Prceedings of the 2019 IEEE Symposium on Security and Privacy (SP)* (2019), 691–706.

[111] Messaoud Mezati and Ines Aouria. 2024. Flink-ML: machine learning in Apache Flink. *Brazilian Journal of Technology* 7, 4 (2024), e74577–e74577.

[112] Paolo Mignone, Gianvito Pio, and Michelangelo Ceci. 2022. Distributed heterogeneous transfer learning for link prediction in the positive unlabeled setting. In *2022 IEEE international conference on big data (Big Data)*. IEEE, 5536–5541.

[113] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In *Proceedings of the International Conference on Machine Learning* (2016), 1928–1937.

[114] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529–533.

[115] 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.

[116] Basil Mustafa, Aaron Loh, Jan Freyberg, Patricia MacWilliams, Megan Wilson, Scott Mayer McKinney, Marcin Sieniek, Jim Winkens, Yuan Liu, Peggy Bui, et al. 2021. Supervised transfer learning at scale for medical imaging. *arXiv preprint arXiv:2101.05913* (2021).

[117] Arun Nair, Praveen Srinivasan, Sam Blackwell, Cagdas Alcicek, Rory Fearon, Alessandro De Maria, Vedavyas Panneershelvam, Mustafa Suleyman, Charles Beattie, Stig Petersen, et al. 2015. Massively parallel methods for deep reinforcement learning. *arXiv preprint arXiv:1507.04296* (2015).

[118] Milad Nasr, Reza Shokri, and Amir Houmansadr. 2019. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE symposium on security and privacy (SP)*. IEEE, 739–753.

[119] Milad Nasr, Shuang Songi, Abhradeep Thakurta, Nicolas Papernot, and Nicholas Carlin. 2021. Adversary instantiation: Lower bounds for differentially private machine learning. In *2021 IEEE Symposium on security and privacy (SP)*. IEEE, 866–882.

[120] Omar Nassef, Wenting Sun, Hakimeh Purmehdi, Mallik Tatipamula, and Toktam Mahmoodi. 2022. A survey: Distributed Machine Learning for 5G and beyond. *Computer Networks* 207 (2022), 108820.

[121] Hung Nguyen, Di Zhuang, Pei-Yuan Wu, and Morris Chang. 2020. Autogan-based dimension reduction for privacy preservation. *Neurocomputing* 384 (2020), 94–103.

[122] Mehdi Noroozi, Ananth Vinjimoor, Paolo Favaro, and Hamed Pirsiavash. 2018. Boosting self-supervised learning via knowledge transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), 9359–9367.

[123] Olga Ohrimenko, Felix Schuster, Cédric Fournet, Aastha Mehta, Sebastian Nowozin, Kapil Vaswani, and Manuel Costa. 2016. Oblivious {Multi-Party} machine learning on trusted processors. In *25th USENIX Security Symposium (USENIX Security 16)*. 619–636.

[124] Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. 2019. Knockoff Nets: Stealing Functionality of Black-Box Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[125] Sinno Jialin Pan and Qiang Yang. 2010. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering* 22, 10 (2010), 1345–1359. https://doi.org/10.1109/TKDE.2009.191

[126] Nicolas Papernot and Thomas Steinke. 2021. Hyperparameter tuning with renyi differential privacy. *arXiv preprint arXiv:2110.03620* (2021).

[127] Jay H Park, Sunghwan Kim, Jinwon Lee, Myeongjae Jeon, and Sam H Noh. 2019. Accelerated training for CNN distributed deep learning through automatic resource-aware layer placement. *arXiv preprint arXiv:1901.05803* (2019).

[128] Sangjoon Park, Gwanghyun Kim, Jeongsol Kim, Boah Kim, and Jong Chul Ye. 2021. Federated split task-agnostic vision transformer for COVID-19 CXR diagnosis. *Advances in Neural Information Processing Systems* 34 (2021), 24617–24630.

[129] Suhyun Park, Hee-Gon Kim, Jibum Hong, Stanislav Lange, Jae-Hyoung Yoo, and James Won-Ki Hong. 2020. Machine learning-based optimal VNF deployment. In *Proceedings of the 2020 21st Asia-Pacific Network Operations and Management Symposium (APNOMS)* (2020), 67–72.

[130] Sangjoon Park and Jong Chul Ye. 2022. Multi-Task Distributed Learning using Vision Transformer with Random Patch Permutation. *arXiv preprint arXiv:2204.03500* (2022).

[131] Dario Pasquini, Giuseppe Ateniese, and Massimo Bernaschi. 2021. Unleashing the tiger: Inference attacks on split learning. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. 2113–2129.

[132] Diego Peteiro-Barral and Bertha Guijarro-Berdiñas. 2013. A survey of methods for distributed machine learning. *Progress in Artificial Intelligence* 2, 1 (2013), 1–11.

[133] Trung V Phan, Nguyen Khac Bao, and Minho Park. 2017. Distributed-SOM: A novel performance bottleneck handler for large-sized software-defined networks under flooding attacks. *Journal of Network and Computer Applications* 91 (2017), 14–25.

[134] Fardin Jalil Piran, Zhiling Chen, Mohsen Imani, and Farhad Imani. 2025. Privacy-preserving federated learning with differentially private hyperdimensional computing. *Computers and Electrical Engineering* 123 (2025), 110261.

[135] Rajashree Manjulalayam Rajendran. 2024. Distributed Computing For Training Large-Scale AI Models in. NET Clusters. *Journal of Computational Intelligence and Robotics* 4, 1 (2024), 64–78.

[136] Amin Rakhsha, Goran Radanovic, Rati Devidze, Xiaojin Zhu, and Adish Singla. 2020. Policy teaching via environment poisoning: Training-time adversarial attacks against reinforcement learning. In *International Conference on Machine Learning*. PMLR, 7974–7984.

[137] Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletari, Holger R Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N Galtier, Bennett A Landman, Klaus Maier-Hein, et al. 2020. The future of digital health with federated learning. *NPJ digital medicine* 3, 1 (2020), 119.

[138] Ronald L Rivest, Len Adleman, Michael L Dertouzos, et al. 1978. On data banks and privacy homomorphisms. *Foundations of secure computation* 4, 11 (1978), 169–180.

[139] Hasim Sak, Oriol Vinyals, Georg Heigold, Andrew Senior, Erik McDermott, Rajat Monga, and Mark Mao. 2014. Sequence discriminative distributed training of long short-term memory recurrent neural networks. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2014).

[140] Salman Salloum, Ruslan Dautov, Xiaojun Chen, Patrick Xiaogang Peng, and Joshua Zhexue Huang. 2016. Big data analytics on Apache Spark. *International Journal of Data Science and Analytics* 1, 3 (2016), 145–164.

[141] Mohammad Reza Samsami and Hossein Alimadad. 2020. Distributed Deep Reinforcement Learning: An Overview. *CoRR* abs/2011.11012 (2020).

[142] Thomas Sandholm, Sayandev Mukherjee, and Bernardo Huberman. 2025. SAFE: secure aggregation with failover and encryption. *ACM Transactions on Modeling and Performance Evaluation of Computing Systems* 10, 1 (2025), 1–28.

[143] Amedeo Sapio, Marco Canini, Chen-Yu Ho, Jacob Nelson, Panos Kalnis, Changhoon Kim, Arvind Krishnamurthy, Masoud Moshref, Dan Ports, and Peter Richtárik. 2021. Scaling distributed machine learning with In-Network Aggregation. In *Proceedings of 18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)* (2021), 785–808.

[144] Jie Shen. 2021. On the power of localized perceptron for label-optimal learning of halfspaces with adversarial noise. In *International Conference on Machine Learning*. PMLR, 9503–9514.

[145] Bichen Shi, Elias Z Tragos, Makbule Gulcin Ozsoy, Ruihai Dong, Neil Hurley, Barry Smyth, and Aonghus Lawlor. 2021. DARES: an asynchronous distributed recommender system using deep reinforcement learning. *IEEE Access* 9 (2021), 83340–83354.

[146] Jiachen Shi, Yi Fan, Guoqiang Zhou, and Jun Shen. 2021. Distributed GAN: Towards A Faster Reinforcement Learning Based Architecture Search. *IEEE Transactions on Artificial Intelligence* (2021).

[147] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP)* (2017), 3–18.

[148] Jinhyun So, Chaoyang He, Chien-Sheng Yang, Songze Li, Qian Yu, Ramy E Ali, Basak Guler, and Salman Avestimehr. 2022. Lightsecagg: a lightweight and versatile design for secure aggregation in federated learning. *Proceedings of Machine Learning and Systems* 4 (2022), 694–720.

[149] Congzheng Song, Thomas Ristenpart, and Vitaly Shmatikov. 2017. Machine learning models that remember too much. In *Proceedings of the 2017 ACM SIGSAC Conference on computer and communications security*. 587–601.

[150] Rafael Stahl, Alexander Hoffman, Daniel Mueller-Gritschneder, Andreas Gerstlauer, and Ulf Schlichtmann. 2021. DeeperThings: Fully distributed CNN inference on resource-constrained edge devices. *International Journal of Parallel Programming* 49, 4 (2021), 600–624.

[151] Phillip Swazinna, Steffen Udluft, and Thomas Runkler. 2021. Overcoming model bias for robust offline deep reinforcement learning. *Engineering Applications of Artificial Intelligence* 104 (2021), 104366.

[152] Anum Talpur and Mohan Gurusamy. 2022. GFCL: A GRU-based Federated Continual Learning Framework against Adversarial Attacks in IoV. *arXiv preprint arXiv:2204.11010* (2022).

[153] Om Dipakbhai Thakkar, Swaroop Ramaswamy, Rajiv Mathews, and Francoise Beaufays. 2021. Understanding unintended memorization in language models under federated learning. In *Proceedings of the Third Workshop on Privacy in Natural Language Processing*. 1–10.

[154] Yuchen Tian, Weizhe Zhang, Andrew Simpson, Yang Liu, and Zoe Lin Jiang. 2021. Defending against data poisoning attacks: from distributed learning to federated learning. *Comput. J.* 66, 3 (2021), 711–726.

[155] Vale Tolpegin, Stacey Truex, Mehmet Emre Gursoy, and Ling Liu. 2020. Data poisoning attacks against federated learning systems. In *Proceedigns of the European Symposium on Research in Computer Security* (2020), 480–501.

[156] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. 2016. Stealing machine learning models via prediction {APIs}. In *25th USENIX security symposium (USENIX Security 16)*. 601–618.

[157] 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.

[158] Daniel Truhn, Soroosh Tayebi Arasteh, Oliver Lester Saldanha, Gustav Müller-Franzes, Firas Khader, Philip Quirke, Nicholas P West, Richard Gray, Gordon GA Hutchins, Jacqueline A James, et al. 2024. Encrypted federated learning for secure decentralized collaboration in cancer image analysis. *Medical image analysis* 92 (2024), 103059.

[159] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* 30 (2017).

[160] Joost Verbraeken, Matthijs Wolting, Jonathan Katzy, Jeroen Kloppenburg, Tim Verbelen, and Jan S Rellermeyer. 2020. A survey on distributed machine learning. *ACM Computing Surveys* 53, 2 (2020), 1–33.

[161] De Wang, Feiping Nie, and Heng Huang. 2014. Large-scale adaptive semi-supervised learning via unified inductive and transductive model. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2014), 482–491.

[162] Xin Wang. 2022. Shared Loss between Generators of GANs. *arXiv preprint arXiv:2211.07234* (2022).

[163] Yinggui Wang, Yuanqing Huang, Jianshu Li, Le Yang, Kai Song, and Lei Wang. 2024. Adaptive Hybrid Masking Strategy for Privacy-Preserving Face Recognition Against Model Inversion Attack. *arXiv preprint arXiv:2403.10558* (2024).

[164] Zhibo Wang, Zhiwei Chang, Jiahui Hu, Xiaoyi Pang, Jiacheng Du, Yongle Chen, and Kui Ren. 2024. Breaking secure aggregation: Label leakage from aggregated gradients in federated learning. In *IEEE INFOCOM 2024-IEEE Conference on Computer Communications*. IEEE, 151–160.

[165] Ziyao Wang, Zheyu Shen, Yexiao He, Guoheng Sun, Hongyi Wang, Lingjuan Lyu, and Ang Li. 2024. Flora: Federated fine-tuning large language models with heterogeneous low-rank adaptations. *arXiv preprint arXiv:2409.05976* (2024).

[166] Zhibo Wang, Mengkai Song, Zhifei Zhang, Yang Song, Qian Wang, and Hairong Qi. 2019. Beyond inferring class representatives: User-level privacy leakage from federated learning. In *IEEE INFOCOM 2019-IEEE conference on computer communications*. IEEE, 2512–2520.

[167] Zhibo Wang, Mengkai Song, Zhifei Zhang, Yang Song, Qian Wang, and Hairong Qi. 2019. Beyond Inferring Class Representatives: User-Level Privacy Leakage From Federated Learning. In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*. 2512–2520. https://doi.org/10.1109/INFOCOM.2019.8737416

[168] Dawei Wei, Junying Zhang, Mohammad Shojafar, Saru Kumari, Ning Xi, and Jianfeng Ma. 2022. Privacy-Aware Multiagent Deep Reinforcement Learning for Task Offloading in VANET. *IEEE Transactions on Intelligent Transportation Systems* (2022).

[169] Wenqi Wei, Ka-Ho Chow, Yanzhao Wu, and Ling Liu. 2023. Demystifying data poisoning attacks in distributed learning as a service. *IEEE Transactions on Services Computing* (2023).

[170] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. 2016. A survey of transfer learning. *Journal of Big data* 3 (2016), 1–40.

[171] Emily Wenger, Josephine Passananti, Arjun Nitin Bhagoji, Yuanshun Yao, Haitao Zheng, and Ben Y Zhao. 2021. Backdoor attacks against deep learning systems in the physical world. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 6206–6215.

[172] Xi Wu, Matthew Fredrikson, Somesh Jha, and Jeffrey F Naughton. 2016. A methodology for formalizing model-inversion attacks. In *2016 IEEE 29th computer security foundations symposium (CSF)*. IEEE, 355–370.

[173] Young Wu, Jeremy McMahan, Xiaojin Zhu, and Qiaomin Xie. 2024. Data poisoning to fake a nash equilibria for markov games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 15979–15987.

[174] Yawen Wu, Dewen Zeng, Zhepeng Wang, Yiyu Shi, and Jingtong Hu. 2022. Distributed contrastive learning for medical image segmentation. *Medical Image Analysis* 81 (2022), 102564.

[175] Geming Xia, Jian Chen, Chaodong Yu, and Jun Ma. 2023. Poisoning attacks in federated learning: A survey. *Ieee Access* 11 (2023), 10708–10722.

[176] Jiacheng Xia, Gaoxiong Zeng, Junxue Zhang, Weiyan Wang, Wei Bai, Junchen Jiang, and Kai Chen. 2019. Rethinking transport layer design for distributed machine learning. In *Proceedings of the 3rd Asia-Pacific Workshop on Networking 2019* (2019), 22–28.

[177] Yun Xie, Peng Li, Jindan Zhang, and Marek R Ogiela. 2021. Differential privacy distributed learning under chaotic quantum particle swarm optimization. *Computing* 103, 3 (2021), 449–472.

[178] Ke Xu, Ziliang Wang, Wei Zheng, Yuhao Ma, Chenglin Wang, Nengxue Jiang, and Cai Cao. 2022. A Centralized-Distributed Transfer Model for Cross-Domain Recommendation Based on Multi-Source Heterogeneous Transfer Learning. In *2022 IEEE International Conference on Data Mining (ICDM)*. 1269–1274. https://doi.org/10.1109/ICDM54844.2022.00166

[179] Weizheng Xu, Youtao Zhang, and Xulong Tang. 2021. Parallelizing DNN Training on GPUs: Challenges and Opportunities. In *Companion Proceedings of the Web Conference 2021* (2021), 174–178.

[180] Biwei Yan, Kun Li, Minghui Xu, Yueyan Dong, Yue Zhang, Zhaochun Ren, and Xiuzhen Cheng. 2024. On protecting the data privacy of large language models (llms): A survey. *arXiv preprint arXiv:2403.05156* (2024).

[181] Xue Yang, Zifeng Liu, Xiaohu Tang, Rongxing Lu, and Bo Liu. 2024. An Efficient and Multi-Private Key Secure Aggregation Scheme for Federated Learning. *IEEE Transactions on Services Computing* (2024).

[182] Zi Ye. 2024. *Mitigating Poisoning Attacks in Decentralized Federated Learning through Moving Target Defense.* Master's thesis. University of Zurich.

[183] Zipeng Ye, Wenjian Luo, Qi Zhou, and Yubo Tang. 2024. High-fidelity gradient inversion in distributed learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 19983–19991.

[184] Zipeng Ye, Wenjian Luo, Qi Zhou, Zhenqian Zhu, Yuhui Shi, and Yan Jia. 2024. Gradient inversion attacks: Impact factors analyses and privacy enhancement. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).

[185] Chen Yu, Hanlin Tang, Cedric Renggli, Simon Kassing, Ankit Singla, Dan Alistarh, Ce Zhang, and Ji Liu. 2019. Distributed Learning over Unreliable Networks. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 7202–7212. https://proceedings.mlr.press/v97/yu19f.html

[186] Da Yu, Huishuai Zhang, Wei Chen, Jian Yin, and Tie-Yan Liu. 2021. How does data augmentation affect privacy in machine learning?. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 10746–10753.

[187] Pengfei Yu and Heng Ji. 2023. Self information update for large language models through mitigating exposure bias. *ArXiv preprint, abs/2305.18582* (2023).

[188] XIE Yueqi, Minghong Fang, and Neil Zhenqiang Gong. 2024. Fedredefense: Defending against model poisoning attacks for federated learning using model update reconstruction error. In *Forty-first International Conference on Machine Learning*.

[189] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329* (2014).

[190] Fadila Zerka, Samir Barakat, Sean Walsh, Marta Bogowicz, Ralph TH Leijenaar, Arthur Jochems, Benjamin Miraglio, David Townend, and Philippe Lambin. 2020. Systematic review of privacy-preserving distributed machine learning from federated databases in health care. *JCO Clinical Cancer Informatics* 4 (2020), 184–200.

[191] Bin Zhang, Cen Chen, and Li Wang. 2020. Privacy-preserving transfer learning via secure maximum mean discrepancy. *arXiv preprint arXiv:2009.11680* (2020).

[192] Chi Zhang, Zhang Xiaoman, Ekanut Sotthiwat, Yanyu Xu, Ping Liu, Liangli Zhen, and Yong Liu. 2023. Generative gradient inversion via over-parameterized networks in federated learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5126–5135.

[193] Huan Zhang, Hongge Chen, Chaowei Xiao, Bo Li, Mingyan Liu, Duane Boning, and Cho-Jui Hsieh. 2020. Robust deep reinforcement learning against adversarial perturbations on state observations. *Advances in Neural Information Processing Systems* 33 (2020), 21024–21037.

[194] Haobo Zhang, Junyuan Hong, Yuyang Deng, Mehrdad Mahdavi, and Jiayu Zhou. 2023. Understanding deep gradient leakage via inversion influence functions. *Advances in neural information processing systems* 36 (2023), 3921–3944.

[195] Ke Zhang, Dingxin Si, Wei Wang, Jiayu Cao, and Yan Zhang. 2021. Transfer learning for distributed intelligence in aerial edge networks. *IEEE Wireless Communications* 28, 5 (2021), 74–81.

[196] Libo Zhang, Bing Duan, Jinlong Li, Zhan'gang Ma, and Xixin Cao. 2024. A tee-based federated privacy protection method: Proposal and implementation. *Applied Sciences* 14, 8 (2024), 3533.

[197] Man Zhang, Yong Zhou, Jiaqi Zhao, Shixiong Xia, Jiaqi Wang, and Zizheng Huang. 2021. Semi-supervised blockwisely architecture search for efficient lightweight generative adversarial network. *Pattern Recognition* 112 (2021), 107794.

[198] Qianqian Zhang, Aidin Ferdowsi, and Walid Saad. 2021. Distributed generative adversarial networks for mmWave channel modeling in wireless UAV networks. In *Proceedings of the ICC 2021-IEEE International Conference on Communications* (2021), 1–6.

[199] Xuezhou Zhang, Yuzhe Ma, Adish Singla, and Xiaojin Zhu. 2020. Adaptive reward-poisoning attacks against reinforcement learning. In *International Conference on Machine Learning*. PMLR, 11225–11234.

[200] Xiaohan Zhang, Shaonan Wang, Nan Lin, Jiajun Zhang, and Chengqing Zong. 2022. Probing word syntactic representations in the brain by a feature elimination method. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 11721–11729.

[201] Yang Zhang, Fuli Feng, Jizhi Zhang, Keqin Bao, Qifan Wang, and Xiangnan He. 2023. Collm: Integrating collaborative embeddings into large language models for recommendation. *arXiv preprint arXiv:2310.19488* (2023).

[202] Yitian Zhang, Hojjat Salehinejad, Joseph Barfett, Errol Colak, and Shahrokh Valaee. 2019. Privacy preserving deep learning with distributed encoders. In *Proceedings of the 2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)* (2019), 1–5.

[203] Zaixi Zhang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. 2022. Fldetector: Defending federated learning against model poisoning attacks via detecting malicious clients. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*. 2545–2555.

[204] Zhikun Zhang, Min Chen, Michael Backes, Yun Shen, and Yang Zhang. 2022. Inference attacks against graph neural networks. In *Proceedings of the 31th USENIX Security Symposium* (2022), 1–18.

[205] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. 2020. idlg: Improved deep leakage from gradients. *arXiv preprint arXiv:2001.02610* (2020).

[206] Xuejun Zhao, Wencan Zhang, Xiaokui Xiao, and Brian Lim. 2021. Exploiting explanations for model inversion attacks. In *Proceedings of the IEEE/CVF international conference on computer vision*. 682–692.

[207] Zhibin Zhao, Qiyang Zhang, Xiaolei Yu, Chuang Sun, Shibin Wang, Ruqiang Yan, and Xuefeng Chen. 2021. Applications of unsupervised deep transfer learning to intelligent fault diagnosis: A survey and comparative study. *IEEE Transactions on Instrumentation and Measurement* 70 (2021), 1–28.

[208] Hongling Zheng, Li Shen, Anke Tang, Yong Luo, Han Hu, Bo Du, and Dacheng Tao. 2023. Learn from model beyond fine-tuning: A survey. *arXiv preprint arXiv:2310.08184* (2023).

[209] Tingting Zhou, Wei Liu, Congyu Zhou, and Leiting Chen. 2018. GAN-based semi-supervised for imbalanced data classification. In *Proceedings of the 2018 4th International Conference on Information Management (ICIM)* (2018), 17–21.

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

[211] Xiaojin Zhu and Andrew B Goldberg. 2009. Introduction to semi-supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 3, 1 (2009), 1–130.

[212] Yaochen Zhu, Liang Wu, Qi Guo, Liangjie Hong, and Jundong Li. 2024. Collaborative large language model for recommender systems. In *Proceedings of the ACM on Web Conference 2024*. 3162–3172.

[213] Zhuangdi Zhu, Kaixiang Lin, Anil K Jain, and Jiayu Zhou. 2023. Transfer learning in deep reinforcement learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).

A Literature Selection

Table 7: Attacks, Affected Knowledge Components, and Countermeasures in Distributed Knowledge Sharing

Attack	Att Ref	Susceptible Model	Attack input	Defense	Def Ref
Adversarial Attacks (AA)	[38]	DSSL, DRL	Adversarial Examples	DP, SA, WS OB, PS, SR	[1, 13, 67, 110, 175]
Adversarial Perturbations (AP)	[193]	DRL	State Observations	DP, FL	[101, 126]
Backdoor Attacks (BA)	[6, 22, 38]	DTL	Trigger Patterns	DP, AT	[119, 171]
Property Inference (DI)	[17, 110]	DUSL, DSSL	Model Updates and attributes that apply to a subset of the	OB, PS, SR	[67, 110, 203]
Data Leakage (DL)	[210]	DTL	Gradients or Intermediate Results	HE, SS, DP	[4, 33, 115]
Data Poisoning (DP)	[12, 78]	DSL, DSSL	Biasing the model's output	OB, PS, SR, DP	[13, 110, 203]
Data Skewing (DS)	[76, 154]	DSL	Data Distribution, Logits	DP, OB, TEE	[1, 123, 188]
Eavesdropping Attacks (EA)	[71, 118]	DUSL	Communication between the Client and Server	DP, DR, EN	[1, 13, 110]
Feature Estimation (FE)	[110, 149]	DSL, DUSL, DSSL	Features and Statistics	OB, PS, SR, HE, DP	[1, 4, 13, 110, 131, 142, 182]
Gradient Disaggregation (GDP)	[192, 194]	DRL	Aggregated Updated, Individual Client Updates	DP, SA	[1, 13]
Gradient Leakage (GL)	[73, 183, 184, 210]	DRL, DSL, DSSL	Shared Gradient Updates	HE, DP, SA, EA, OB, TE, PS	[1, 5, 13, 123, 148]
Graph Reconstruction (GR)	[26, 65, 204]	DSSL	Outputs or Embeddings	WS, OB, PS, SR	[42, 46, 110, 195]
Inference Attacks (IA)	[110, 147, 204]	DSL, DUSL, DRL	Shared Model Updates, Model's output or Gradient patterns	HE, DP, SA, OB, TE, EN, DA	[43, 62, 82, 83, 106, 158, 186]
Interception during Map Merging (IM)	[48, 95]	DUSL	Partial maps or Model fragments	DP, DR, EN, DA	[20, 49, 96, 121, 137]
Knowledge Transfer Inference (KTI)	[41, 56, 63]	DTL	Embeddings, Victim's dataset poperties	HE, SS, DP	[5, 8, 13, 47]
MMD Privacy Leaks (PL)	[106]	DTL	Statistical features like Maximum Mean Discrepancy	HE, SS, DP	[5, 8, 13, 47]
Memory Corruption (MC)	[108]	DRL	Logits	DP, SA, AT	[1, 13, 107, 118]
Model Inversion (MV)	[172, 194, 206]	DSL	Representative Inputs	HE, DP, SA, EA, OB, TEE	[89, 134, 163, 164, 181, 196]
Model Memorization (MM)	[153]	DSL	Training data updates, Regurgitating Sequences	HE, DP, SA, EA, OB, TEE	[89, 134, 163, 164, 181, 196]
Model Poisoning (MP)	[17, 118, 149]	DUSL, DSSL, DTL	Misclassify inputs	DP, SR, PS, WS, DA	[14, 46, 144, 177, 200]
Model Reconstruction (MR)	[66, 156, 166]	DUSL, DSSL	shared gradients or model updates	DP, SR, PS, WS, DA	[14, 46, 144, 177, 200]

Continued on next page

Table 7 Continued from previous page

Attack	Att Ref	Susceptible Model	Attack input	Defense	Def Ref
Packet Loss (PL)	[99, 185]	DSL	dropping or withholding packets in transit. Lost model updates	HE, DP, SA, EA, OB, TEE	[89, 134, 163, 164, 181, 196]
Parameter Inference (PI)	[90, 124, 167]	DSSL	global model updates model's parameters or architecture	WS, OB, PS, SR	[42, 46, 110, 195]
Poisoned Updates (PU)	[10, 42]	DSL,DTL	model updates	HE, DP, SA, EA, OB, TEE	[89, 134, 163, 164, 181, 196]
Policy Value Manipulation (PV)	[11, 136, 199]	DRL	policy values state inputs or intermediate signals	DP, SA, AT	[1, 13, 107, 118]
Preference Poisoning (PP)	[24, 37, 94]	DUSL	skews the learned reward	DP, DR, EN, DA	[20, 49, 96, 121, 137]
Privacy Re-identification (PR)	[110, 118, 157]	DUSL	observed model outputs/update , shared model parameters	DP, DR, EN, DA	[20, 49, 96, 121, 137]
Recursive Reconstruction (RR)	[47, 194, 205]	DRL	layer-wise gradients, intermediate layers	DP, SA, AT	[1, 13, 107, 118]
Reward Manipulation (RM)	[11, 136, 173]	DRL	reward signal	DP, SA, AT	[1, 13, 107, 118]

B Abbreviations

Table 8: Abbreviations in Distributed Learning, Architectures, Knowledge Components, and Security

Abbreviation	Full Form
General Concepts	
AI	Artificial Intelligence
ML	Machine Learning
DL	Distributed Learning
FL	Federated Learning
IoT	Internet of Things
GPU	Graphics Processing Unit
RDD	Resilient Distributed Dataset
DL Paradigms	
DSL	Distributed Supervised Learning
DUSL	Distributed Unsupervised Learning
DSSL	Distributed Semi-Supervised Learning
DRL	Distributed Reinforcement Learning
DTL	Distributed Transfer Learning
DLLM	Decentralized Large Language Models
Architectures & Models	
MLP	Multilayer Perceptron
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network

Continued on next page

Table 8 – continued from previous page

Abbreviation	Full Form
LSTM	Long Short-Term Memory
GRU	Gated Recurrent Unit
GAN	Generative Adversarial Network
SOM	Self-Organizing Map
AE	Autoencoder
DSSGAN	Distributed Semi-Supervised GAN
DCL	Distributed Contrastive Learning
PPDTL	Privacy Preserving Distributed Transfer Learning
DISHTRA	Distributed Heterogeneous Transfer Learning
PPDUSTR	Privacy Preserving Unsupervised Transfer Learning
PPDESTR	Privacy Preserving Distributed Semi-Supervised Transfer Learning
TF	TensorFlow
TFX	TensorFlow Extended
ETL	Extract, Transform, Load
Knowledge Components	
GL	Gradient Leakage
GDP	Gradient Disaggregation
PU	Poisoned Updates
PL	Packet Loss / Privacy Leak
PD	Parameter Distribution
AP	Aggregation Parameters
LO	Logits
CP	Control Parameters
BS	Batch Size
EF	Error Feedback
LD	Latent Distribution
CS	Cell State
HS	Hidden State
PPt	Partitioning Points
SF	Skewness Factor
MCL	Merged Classification Logits
DRS	Data Representation Snippets
TDM	Tangents of Data Manifold
TP	Task-specific Parameters
IR	Intermediate Representation
SAp	State-Action Pairs
RB	Replay Buffer
Attacks	
MM	Model Memorization
GL	Gradient Leakage

Continued on next page

Table 8 – continued from previous page

Abbreviation	Full Form
MI	Membership Inference
PI	Property Inference
PR	Privacy Re-identification
MP	Model Poisoning
MR	Model Reconstruction
FE	Feature Estimation
IA	Inference Attacks
IM	Interception during Map Merging
DS	Data Skewing
AA	Adversarial Attack
BA	Backdoor Attack
DI	Data Inference
APert	Adversarial Perturbation
KTI	Knowledge Transfer Inference
MC	Memory Corruption
RM	Reward Manipulation
PV	Policy Value Manipulation
RR	Recursive Reconstruction
Defenses	
DP	Differential Privacy
HE	Homomorphic Encryption
SA	Secure Aggregation
EA	Encryption-based Aggregation
TEE	Trusted Execution Environment
OB	Obfuscation
DR	Dimensionality Reduction
EN	Encryption
DA	Data Augmentation
WS	Weighted Steiner Tree
PS	Privacy-preserving Embedding Sharing
SR	Secure Representation Sharing
SS	Secret Sharing
AT	Adversarial Training