Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1, 564-594
2024
Publisher: Learning Gate
DOI: 10.55214/25768484.v9i1.4184

The application of artificial intelligence and machine learning in civil law protection of privacy rights

Weiwang Huang[1*], Ming Hsun Hsieh[2]
[1]Institute of Law, International College, Krirk University, 3 Ram Inthra Rd, Anusawari, Bang Khen, Bangkok 10220, Thailand; y5708211@gmail.com (W.H.).

Abstract: Machine learning has emerged as a core technology in domains such as big data, the Internet of Things (IoT), and cloud computing. The training of machine learning models typically requires extensive datasets, often gathered through crowdsourcing methods. These datasets frequently contain significant amounts of private information, including personally identifiable information (e.g., phone numbers, identification numbers) and sensitive data (e.g., financial, medical, and health records). The efficient and cost-effective protection of such data represents a pressing challenge. This article introduces machine learning and explores the concepts and threats associated with privacy within this context. It focuses on mainstream techniques for privacy protection in machine learning, outlining their underlying mechanisms and distinctive features. The discussion is organised around key frameworks such as differential privacy, homomorphic encryption, and secure multi-party computation, presenting a comprehensive review of recent advancements in the field. A comparative analysis of the strengths and limitations of these mechanisms is provided. Finally, the article examines the developmental trajectory of privacy protection in machine learning and proposes potential future research directions in this critical area.

Keywords: *Data protection, Law, Privacy rights.*

1. Introduction

In recent years, machine learning (ML) has experienced rapid development, emerging as a cornerstone in fields such as image processing, speech recognition, and cyber security. Simultaneously, advancements in computing, storage, and networking technologies have led to an exponential increase in data volumes [1, 2]. Additionally, media such as the Internet of Things (IoT), social media, and smart phones produce vast amounts of data every minute. Data holders can transmit this information to cloud service providers (CSPs) to identify potential data models that may support decision-making, improve business processes, and offer value-added services such as prediction and recommendation systems.

Against this backdrop, many CSPs have introduced Machine Learning as a Service (MLaaS). MLaaS provides automated solutions for data processing, model training, prediction services, and deployment, enabling machine learning practitioners to deploy applications on cloud platforms without needing to establish their own large-scale infrastructure or computational resources. Prominent MLaaS platforms include Google Prediction API [3] Amazon ML [4] Microsoft Azure ML [5] and BigML [6]. A typical cloud-based machine learning architecture is illustrated in Figure 1.

* Correspondence: y5708211@gmail.com

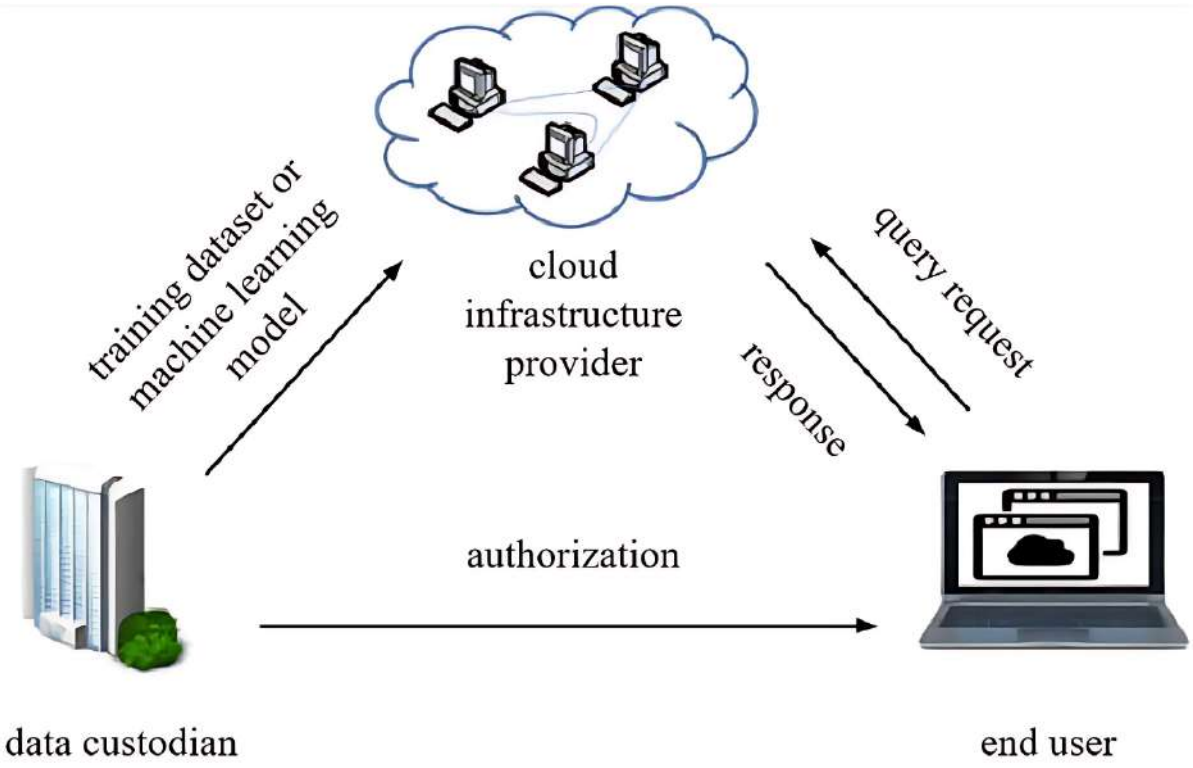

Figure 1.
Architecture and privacy threat model of cloud-based machine learning.

In this architecture, CSPs may represent third-party MLaaS platforms, partner organisations, or in-house applications operated at remote or independent facilities. Data holders, such as governments, banks, hospitals, insurance companies, or e-commerce websites, can store, process, or utilise services provided by cloud platforms. End users, such as business employees, doctors, or clinical staff, interact with the deployed services by uploading prediction requests to the CSP, which then returns the model's prediction results.

While MLaaS offers compelling benefits, it also introduces significant concerns regarding data security and privacy, as depicted in Figure 1 [7]. During the training phase, a malicious CSP can make minor alterations to the training algorithm to generate a high-quality model that satisfies standard ML metrics (e.g., accuracy and general ability) while extracting detailed information about the training data [8]. Even if a malicious CSP lacks direct access to the dataset, sensitive information can still be inferred from the model's parameters.

Similarly, privacy breaches can occur during the prediction phase. Recent studies have begun to address these privacy issues [9-11]. For instance, clients are required to upload pre-trained models to the CSP to enable prediction services. However, model leakage can result in the loss of proprietary data, potentially compromising the original dataset. Furthermore, even with black-box access, a malicious remote user can infer sensitive training data through carefully designed queries and model outputs [12-17]. These privacy concerns represent a significant challenge to the advancement of cloud-based machine learning.

At the same time, privacy, as a fundamental human right, is of paramount importance to both individuals and organisations. The growing emphasis on data privacy and security has become a global trend. The European Union's General Data Protection Regulation (GDPR) [18] which came into effect on 25 May 2018, mandates that data processing must be based on explicit user consent and grants users the "right to be forgotten," allowing them to delete or withdraw personal data at any time. Similarly,

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 564-594, 2025
DOI: 10.55214/25768484.v9i1.4184

the California Consumer Privacy Act (CCPA) [19] regarded as the strictest privacy law in the United States, came into effect on 1 January 2020, aiming to enhance consumer privacy rights and data protection. Non-compliant entities face severe penalties. In China, the Cyber security Law, implemented in June 2017 [20] prohibits individuals and organisations from illegally obtaining personal information or disclosing it to third parties without the subject's consent. These regulations pose new challenges to traditional data-processing models in artificial intelligence.

This paper begins by introducing foundational knowledge on privacy protection in machine learning, including an overview of ML, definitions of ML privacy, adversarial models, and privacy protection scenarios (Section 1). It then examines typical privacy threats in machine learning and categorises existing privacy-preserving solutions. Subsequently, the study explores various mechanisms for privacy preservation in ML, analysing key concepts, representative methods, and their application scenarios for each category. Finally, the paper summarises the findings and discusses potential research directions and future trends in this domain.

2. Methodology

2.1. Definition of Machine Learning

Machine learning (ML) is an interdisciplinary research field encompassing computer science, probability and statistics, psychology, and neuroscience. It enables computers to emulate human learning by analysing existing data to generate useful models, which in turn support decision-making and predictions about future actions. Based on the characteristics of the data used for learning, ML is typically categorised into supervised learning, semi-supervised learning, unsupervised learning, and reinforcement learning.

The process of solving problems with ML involves two main stages: training and prediction. During training, a target model is developed, which is later used for prediction. In supervised learning, for example, the ML model is a parametric function $f(\theta):X\rightarrow Y$, mapping input data $x\in X$ (features) to output data $y\in Y$ (labels). In classification tasks, X represents a d-dimensional vector space, while Y comprises a set of discrete classes. The training process seeks optimal parameters $\theta\backslash theta\theta$ that accurately reflect the relationship between X and Y, allowing new data to be classified correctly. Given a dataset with N training samples, the loss function ℓ, as shown in Equation (1), measures the error between actual and predicted outputs. The goal of training is to minimise this loss function to derive the optimal model parameters, θ^*:

$$\theta^* = \text{argmin}_\theta \Omega(\theta) + \frac{1}{N}\sum_{i=1}^{N} \ell\left(y_i, f_\theta(x_i)\right) \quad (1)$$

Here, $\Omega(\theta)$ represents a regularisation term to prevent overfitting. Depending on whether the training data is collected centrally beforehand, ML training methods can be divided into centralised learning, distributed learning, and federated learning.

2.2. Privacy in Machine Learning

Privacy is a multifaceted concept without a universally accepted definition. A seminal work, *The Right to Privacy*, published in the *Harvard Law Review* in 1890 [21] defined privacy as "the right to be left alone." The 1966 United Nations International Covenant on Civil and Political Rights [22] described privacy as the right to protection against arbitrary or unlawful interference with an individual's personal life, family, home, or correspondence, and attacks on honour or reputation. Other scholars, such as Saltzer and Schroeder [23] have defined privacy as the ability of individuals or organisations to control the disclosure of their information. Zhou, et al. [24] define privacy as "sensitive data or characteristics represented by the data that the data owner is unwilling to disclose." In ML, privacy can be categorised into three primary aspects: training data privacy, model privacy, and prediction output privacy.

- **Training Data Privacy**: This involves protecting users' personally identifiable information (PII) and sensitive data. PII includes unique identifiers such as names, ID numbers, phone numbers, and

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 564-594, 2025
DOI: 10.55214/25768484.v9i1.4184

email addresses, while quasi-identifiers refer to combinations of attributes that can identify individuals, such as address, gender, and date of birth. Sensitive information encompasses demographic details, financial records, health data, and daily activities.

- **Model Privacy**: This refers to safeguarding proprietary information related to ML models, including training algorithms, model topology, parameters, activation functions, and hyperparameters. For instance, as shown in the encrypted prediction as a service (EPAAS) architecture [25] ML models belong to service providers and are only accessible to authorised users. However, adversaries may engage in model extraction attacks to steal models for malicious purposes.
- **Prediction Output Privacy**: Prediction result privacy refers to sensitive information returned by a machine learning model in response to a user's prediction input request, which the user does not wish to disclose. For instance, model prediction results may include a user's medical diagnosis, such as the probability of having a specific disease. This information constitutes personal privacy for the user, but untrusted service providers or third parties could potentially intercept or misuse it. Xie, et al. [26] proposed the privacy-preserving neural network model *Crypto-Nets*, which employs homomorphic encryption techniques to process encrypted data directly. This model generates predictions on ciphertext and returns encrypted prediction results, thereby ensuring the privacy of online medical diagnostic predictions. The protection of training data privacy, model privacy, and prediction result privacy is critical when using machine learning. Any compromise of this information could jeopardise the security of users' sensitive data or result in significant economic losses for service providers. These concerns represent a substantial challenge to the advancement of cloud computing. Consequently, machine learning service systems based on cloud computing must prioritise privacy issues and continuously enhance their privacy protection capabilities.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 564-594, 2025
DOI: 10.55214/25768484.v9i1.4184

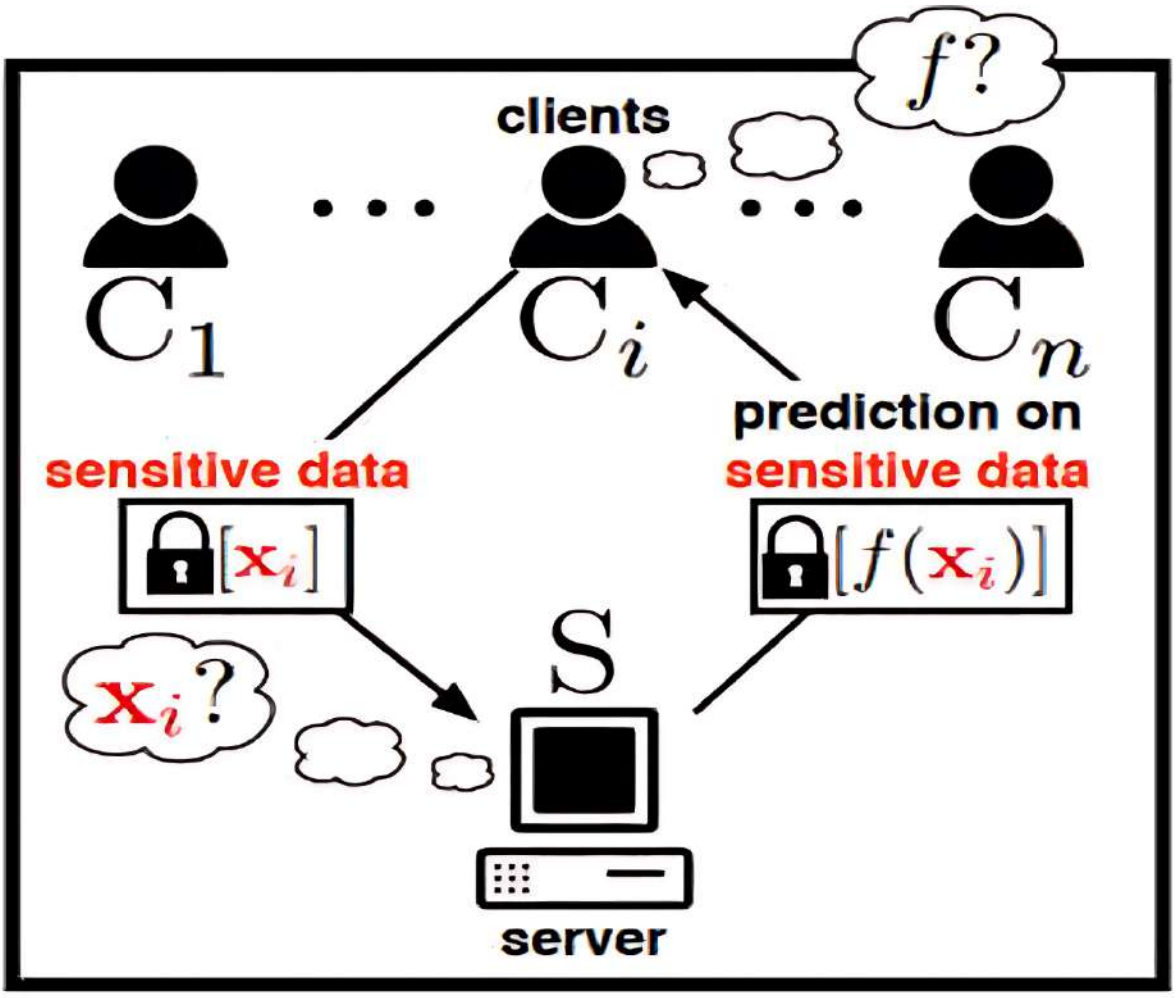

Figure 2.
Architecture of EPAAS.

2.3. Adversarial Models in ML Privacy Attacks

Adversarial models in ML privacy attacks are characterised by their goals, knowledge, capabilities, and strategies. Table 1 summarises these elements:

Table 1.
Adversarial model of privacy attack.

Adversary goals	Adversary knowledge	Adversary capabilities	Adversary strategies
Training data privacy, model privacy, prediction output privacy	White-box: full model knowledge	Strong: participate in training or access data	Model inversion, extraction, and membership inference attacks
	Black-box: no model knowledge	Weak: access limited outputs	

Adversarial attacks on machine learning (ML) models aim to compromise the confidentiality of these systems, focusing on accessing sensitive information such as training data privacy, model parameters, and prediction outcomes. The primary objective of adversaries in privacy attacks is to breach the confidentiality of the ML model. A well-designed ML system must ensure that critical information remains inaccessible to unauthorised users. For instance, an ML-based medical diagnostic

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 564-594, 2025
DOI: 10.55214/25768484.v9i1.4184

system should prevent adversaries from analysing the model to recover patient information [27]. Similarly, when the model itself represents intellectual property, such as in financial market systems, its structure and parameters must remain confidential [28].

Adversarial Knowledge refers to the information an adversary possesses about the target model and its operational environment. This may include the distribution of the training dataset, the model's structure and parameters, and its decision functions. Depending on the extent of the adversary's knowledge, attacks can be categorised as white-box or black-box. In white-box attacks, the adversary has access to information about the model or its training data, such as the ML model's architecture, parameters, or parts of the training dataset. Conversely, black-box attacks assume the adversary has no prior knowledge of the model; instead, they infer model information by exploiting its vulnerabilities and analysing outputs generated from a series of carefully designed inputs [29].

Adversarial Capabilities denote the methods and resources available to an adversary. During the data collection phase, adversaries may directly access user data. In the training phase, their capabilities could include interfering with the training process, accessing training data, or collecting intermediate outputs. During the prediction phase, adversaries might exploit access to the model to retrieve information about training data. Depending on the level of control and destructive potential, adversaries can be classified as strong or weak. Strong adversaries possess capabilities such as participating in model training or accessing model and training data, while weak adversaries rely on indirect methods to gather information about the model's features [30].

Adversarial Strategies encompass the specific approaches adversaries employ to achieve their objectives, which are determined by a combination of their goals, knowledge, and capabilities. During the data collection phase, adversaries may directly access data. However, in the training and prediction phases, their strategies can be divided into direct and indirect attacks. Direct attacks involve extracting training data information or determining whether a particular data point was included in the training dataset based on model prediction outputs. Indirect attacks involve first compromising model parameters to construct a substitute model, which is then used to infer information about the training dataset.

Specific adversarial strategies include model inversion attacks, model extraction attacks, and membership inference attacks. Among these, model inversion and membership inference attacks are classified as direct strategies, while model extraction attacks fall under indirect strategies.

2.4. Privacy Protection Scenarios and Typical Privacy Threats in Machine Learning

Privacy protection scenarios refer to specific contexts in machine learning (ML) where privacy breaches might occur, necessitating appropriate countermeasures. Different privacy-preserving techniques are tailored to distinct scenarios, and understanding these scenarios is essential for designing effective privacy protection strategies. Factors such as the ML stage, model training methods, data distribution, and the trustworthiness of participating entities determine these scenarios. Typical privacy protection scenarios include the data collection phase in centralised learning [31] the model training phase in federated learning [32] prediction services in patient-centred online healthcare systems [30], and online services in cloud-based financial systems [28]. This section focuses on the first two scenarios.

2.4.1. Centralised Learning

The defining characteristic of centralised learning is its simplicity in system deployment, as it eliminates the need for multi-node deployment or distributed collaboration. In this setup, a central server handles the entire ML process, including data collection, model training, model publishing, and prediction. Each of these stages presents potential privacy risks.

During the data collection phase, the lack of unified standards often results in untrustworthy data collectors excessively gathering and selling user data, making the theft of raw user data one of the most critical privacy challenges in ML systems. To mitigate such risks, companies like Apple and Google have adopted local differential privacy (LDP) techniques to safeguard user data during collection [31].

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 564-594, 2025
DOI: 10.55214/25768484.v9i1.4184

2.4.2. Federated Learning

Federated learning, which avoids centralised storage of user data, offers significant privacy benefits. However, the model training phase in federated learning is susceptible to various malicious attacks. Research by Nasr, et al. [33] demonstrates that a curious parameter server or even a participant can launch highly effective membership inference attacks against other participants. For example, in experiments using the DenseNet model on the CIFAR100 dataset, a curious central parameter server achieved a membership inference accuracy of 79.2% by analysing parameter updates from all participants. Similarly, local participants observing aggregated parameter updates achieved a 72.2% accuracy.

Adversaries can also exploit stochastic gradient descent (SGD) to extract additional information from participants' training data. By isolating participants during parameter updates, a central server increased the accuracy of active inference attacks on the DenseNet model to 87.3%. Malicious participants can infer other participants' training data by monitoring global parameter changes and their adversarial parameter updates. Consequently, the model training phase is a critical privacy protection scenario in federated learning.

2.4.3. Machine Learning as a Service (MLaaS)

MLaaS systems typically involve two stages: model training and prediction services. Both stages are vulnerable to three main types of privacy attacks: model inversion attacks, model extraction attacks, and membership inference attacks. Table 2 summarises these typical threats across the ML lifecycle.

Table 2.
Typical privacy threats in machine learning.

Stage	Model inversion attacks	Model extraction attacks	Membership inference attacks
Training	[7, 8, 27, 34]	[7]	
Prediction	[12-17, 35, 36]	[37]	[36, 38]

2.4.4. Model Inversion Attack

Model inversion attacks involve extracting information about the training data from the model's prediction outputs [12]. This technique becomes particularly effective when combined with generative adversarial networks (GANs). For instance, Fredrikson, et al. [13] demonstrated a reverse attack on a linear regression-based personalised medicine system, exposing patient privacy and potentially misconfiguring medications, which could endanger lives. In another study, Fredrikson, et al. [12] reconstructed facial images using a neural network-based facial recognition model. Similarly, Hitaj, et al. [34] showed that distributed or federated learning structures struggle to protect honest participants' training datasets from GAN-based attacks.

2.4.5. Model Extraction Attack

Model extraction attacks involve adversaries gaining black-box access to a target model and attempting to retrieve its parameters or structure, or even reconstructing a model equivalent to the target Tramèr, et al. [37]. Song, et al. [7] revealed that malicious ML algorithms could create high-quality models that meet accuracy and generalisation requirements while leaking substantial information about the training data. Florian [37] further demonstrated that adversaries could extract model information by querying the prediction service API a limited number of times. For an N-dimensional linear model, they showed that just N+1 queries could theoretically suffice to compromise the model.

2.4.6. Membership Inference Attack

Membership inference attacks allow adversaries to determine whether a specific data point was part of the training dataset by accessing the model's prediction API and analysing the confidence of predictions [36]. This attack is particularly effective against overfitted models. Shokri, et al. [36] used membership inference attacks to infer whether a given data point was included in a training dataset.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 564-594, 2025
DOI: 10.55214/25768484.v9i1.4184

Melis, et al. [38] demonstrated that in collaborative and federated learning scenarios, adversaries could infer not only the presence of specific data points in other participants' training datasets but also attributes and temporal changes in the training data.

2.4.7. Classification of Privacy-Preserving Techniques in Machine Learning

Privacy protection methods in ML can be categorised based on the type of ML model, the ML process, the training methodology, and the specific privacy-preserving techniques employed. Table 3 summarises these classifications.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 564-594, 2025
DOI: 10.55214/25768484.v9i1.4184

Table 3.
Classification of privacy protection techniques in machine learning.

Categories			Type of machine learning model			
By Type of Machine Learning Model	Supervised learning privacy protection	Linear regression	Ref [13] Ref [39]			
		Logistic regression	Ref [40]			
		Support vector machines	Ref [9]			
		Decision trees and random forests	Ref [41]			
		Extreme learning	Ref [42]			
		Bayesian algorithms	Ref [43] Ref [44]			
		Neural networks	DP-GANs [45] Ref [46] [47] CryptoDL	AdLM [48] crypto-nets [26] pCDBN [49]	Ref [50] CryptoNets [51] LPP-CNN [52]	Ref [53] Ref [54] OPSR [55]
	Semi-supervised learning privacy protection		PATE [56]			
	Unsupervised learning privacy protection	k-Means	Ref [57]	Ref [58]		
	Reinforcement learning privacy protection	Q-learning	LiPSG [59]			
By Stage of the Machine Learning Process	Training Phase Privacy Protection		Ref [44] PPDL [60]	Ref [53] Ref [61]	Ref [46]	Ref [62]
	Prediction Phase Privacy Protection		Ref [9] Ref [55] Rf [68]	Ref [43] CryptoDL [56] Ref [69]	crypto-nets [26] TAPAS [29] FHE-DiNN [70]	CryptoNets [54] Ref [67]
By Model Training Approach	Centralised Learning Privacy Protection		Ref [9]	Ref [53]	Ref [64]	Ref [66]
	Distributed Learning Privacy Protection		Ref [21] Ref [61]	Ref [43] Ref [71]	Ref [45] Ref [72]	Ref [46] Ref [73]
	Federated Learning Privacy Protection		Ref [22]	Ref [23]	Ref [24]	Ref [73]
By Privacy Protection Technique	Differential Privacy	Input Perturbation	DP-GANs [49]	DPGAN [74]	Ref [75]	
		Intermediate Parameter Perturbation	AdLM [50]	Ref [36]	Ref [76]	Ref [77]
		Target Perturbation	Ref [44]	dPAs [78]	pCDBN [57]	Ref [51]
		Output Perturbation	Ref [73]	PATE [60]	Ref79]	
	Homomorphic Encryption	Without Polynomial Approximation	Ref [9] Ref [53] Ref [69]	Ref [47] TAPAS [29] FHE-DiNN [70]	Ref [48] Ref [67] Ref [80]	Ref [52] Ref [68] Ref [81]
		With Polynomial Approximation	crypto-nets [30] Ref [64]	[54] CryptoNets PPDL [65]	Ref [55] Ref [66]	CryptoDL [56]
	Secure Multi-party Computation	Traditional Distributed Learning	Ref [45]] Ref [72] Ref [85]	Ref [46] Ref [82] Ref [62]	Ref [61] Ref [83]	Ref [71] Ref [84]
		Based on 2PC Architecture	Ref [39] EzPC [63] LPP-CNN [52]	SecureML [64] Chameleon [65] POR [66]	DeepSecure [67] GAZELLE [68] OPSR [55]	MiniONN [69] TASTY [70] LiPSG [59]

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 564-594, 2025
DOI: 10.55214/25768484.v9i1.4184

3. Results and Discussion

3.1. Machine Learning Privacy Protection Mechanisms Based on Differential Privacy

Differential privacy (DP) is a widely recognised and rigorous privacy protection technique, first proposed by Dwork [71]. The DP framework ensures that even when a malicious adversary gains access to the published results of a dataset, they cannot infer sensitive information about individual users. Applying DP to machine learning (ML) models offers the capability to protect training data against reverse engineering attacks when model parameters are released. As a result, numerous studies have incorporated DP into ML models.

Privacy Protection Mechanisms Based on Input Perturbation: Input perturbation refers to the application of random noise to training data prior to its exposure to the model. This technique prevents the model from accessing raw user data. By protecting the data before the training process begins, input perturbation significantly reduces the risk of sensitive information leakage. From a privacy perspective, this method is considered more reliable than perturbations introduced at later stages. In existing literature, two primary methods of input perturbation are commonly employed: differential privacy data synthesis and localised differential privacy perturbation.

- Differential Privacy Data Synthesis
  Differential privacy data synthesis can be regarded as a preprocessing step for training data. This approach generates artificially synthesised data that shares statistical characteristics and format with the original input data, thereby achieving privacy protection for the original data.
- Localised Differential Privacy Perturbation
  Localised differential privacy models focus on protecting the privacy of communications between individuals and untrusted servers. In this framework, each user applies differential privacy perturbation locally to their original data before transmitting the processed data to the data collector [72].

Advances in Privacy-Preserving Models Using Generative Adversarial Networks: In recent years, generative adversarial networks (GANs) and their variants have effectively addressed the issue of data scarcity as generative models. However, GANs pose a risk of disclosing private training data. To address this limitation, Beaulieu-Jones, et al. [45] proposed a model called DP-GANs, which employs DP-SGD (Differentially Private Stochastic Gradient Descent) to train auxiliary classifier generative adversarial networks (AC-GANs). This model utilises deep neural networks to generate synthetic data under differential privacy constraints, providing a solution for sharing clinical research data while preserving patient privacy.

As illustrated in Figure 4, the DP-GAN model incorporates two neural networks:

- The Generator (G) is trained to produce new data x′x'x′ that closely resembles the original data xxx using a set of random inputs zzz.
- The Discriminator (D) is designed to distinguish between real samples and those generated by the generator.
- The model's value function is expressed as follows:

$$\min_G \max_D V(G,D) = \mathbb{E}_{x\sim p_{daa}(x)}\left[\log\left(D(x)\right)\right] + \mathbb{E}_{z\sim p_z(z)}\left[\log\left(1D\left(G(z)\right)\right)\right] \quad (2)$$

This equation constructs a mini max game between the two networks, achieving Nash equilibrium through adversarial training. During the training process, (ε, δ)-differential privacy protection is applied to the discriminator's gradients. Due to the immunity of differential privacy to post-processing [73] the generator also inherits (ε, δ)-differential privacy protection. Furthermore, within the DP-GAN framework, the discriminator is the only component granted access to the real, private data. Consequently, even if an adversary gains access to the generator, they cannot extract private information about the training data.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 564-594, 2025
DOI: 10.55214/25768484.v9i1.4184

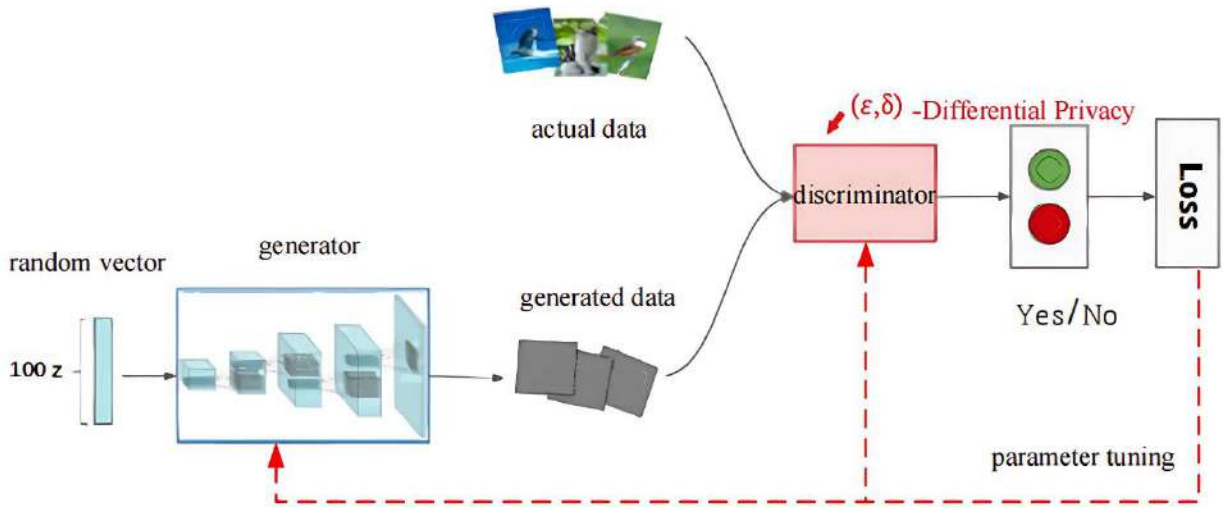

Figure 3.
Frame work for DP-GANs.

Addressing the issues of training instability, gradient vanishing, and lack of diversity associated with the use of GANs, as highlighted in Beaulieu-Jones, et al. [45] and Xie, et al. [74] proposed a differentially private generative adversarial network (DPGAN). This model is built upon the Wasserstein GAN (WGAN) framework, which formulates another two-player minimax game. Unlike the KL divergence and JS divergence employed in traditional GANs, the Wasserstein distance is advantageous as it remains effective even when two distributions do not overlap, thereby mitigating issues such as training instability and gradient vanishing. During training, the DPGAN model employs (ε, δ)-differential privacy to safeguard the training data. The Moments Accountant mechanism is utilized to precisely control the privacy loss during the training process, ensuring the usability of the model.

To address the decline in utility of the generated data, Bindschaedler, et al. [75] introduced a formal privacy guarantee for the publication of high-dimensional sensitive data, using the plausible deniability standard [76] to measure the privacy of the generated data. Mechanisms that satisfy the plausible deniability standard consist of two independent modules: a generation module and a privacy testing module. This method generates data and then releases only a subset that satisfies privacy requirements, enabling the creation of highly practical synthetic data. Differential privacy is achieved by applying a privacy testing mechanism to reject undesirable samples, thereby ensuring plausible deniability. The (k,γ)(k, \gamma)(k,γ)-plausible deniability mechanism, defined in Equation (7), ensures input indistinguishability. This implies that, by observing the output set (i.e., the generated data), an adversary cannot determine whether a particular data record is part of the input set (i.e., the original data). A larger value of privacy parameter k indicates a larger indistinguishable input data set, while a privacy parameter γ closer to 1 signifies stronger indistinguishing ability of input data records.

$$\gamma^{-1} \leq \frac{\Pr(d_i)y=M(d_i))}{\Pr(d_i)y=M(d_j))} \leq \gamma \quad (3)$$

where ∀i,j∈(1,2,…,k); d_i represents the original input data; M(d) denotes the probabilistic generative model; y is the generated data; and k, γ are the privacy parameters.

Privacy Protection Schemes Based on Intermediate Parameter Perturbation: Intermediate parameter perturbation involves adding Laplace or Gaussian noise to gradient or feature parameters during model training to prevent adversaries from extracting private information about the model or training data. Recent research has introduced innovative improvements, such as more precise noise addition and stricter measurement of privacy loss, which hold significant value for model optimisation.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 564-594, 2025
DOI: 10.55214/25768484.v9i1.4184

In response to the potential privacy leakage caused by directly sharing training datasets in deep learning, Shokri and Shmatikov [32] proposed a distributed selective stochastic gradient descent algorithm (DSSGD). This algorithm enables multiple parties to collaboratively train an accurate target model through parallel asynchronous training without sharing raw training data. The DSSGD framework, illustrated in Figure 5, updates server parameters using the rule shown in Equation (8). Here, α is the learning rate, Wglobal represents the global parameters of the central server, which are broadcast to all participants for download and update, and the vector G contains approximately 1%-10% of the gradient parameters from each participant. To ensure that parameter updates do not reveal excessive information about the training dataset, the algorithm incorporates ϵ differential privacy noise (Laplace noise) into the gradient parameters. Participants do not need to interact with one another, as they train locally using their individual models. Experimental results demonstrate that, for a large number of participants, the accuracy of the collaboratively trained model surpasses that of independently trained models when participants share a substantial portion of their gradients.

$$W_{global} \leftarrow W_{global} - \alpha G_{local}^{selectine} \quad (4)$$

Building on the work of Shokri and Shmatikov [32] and Liu, et al. [77] proposed a privacy-preserving collaborative deep learning system for mobile environments that does not share local raw data. This system enables multiple sites to train deep learning models by sharing only a portion of the parameters. Mobile devices train locally on their respective datasets and upload the trained parameters to the global server (XMPP) using an iterative and asynchronous parameter exchange protocol.

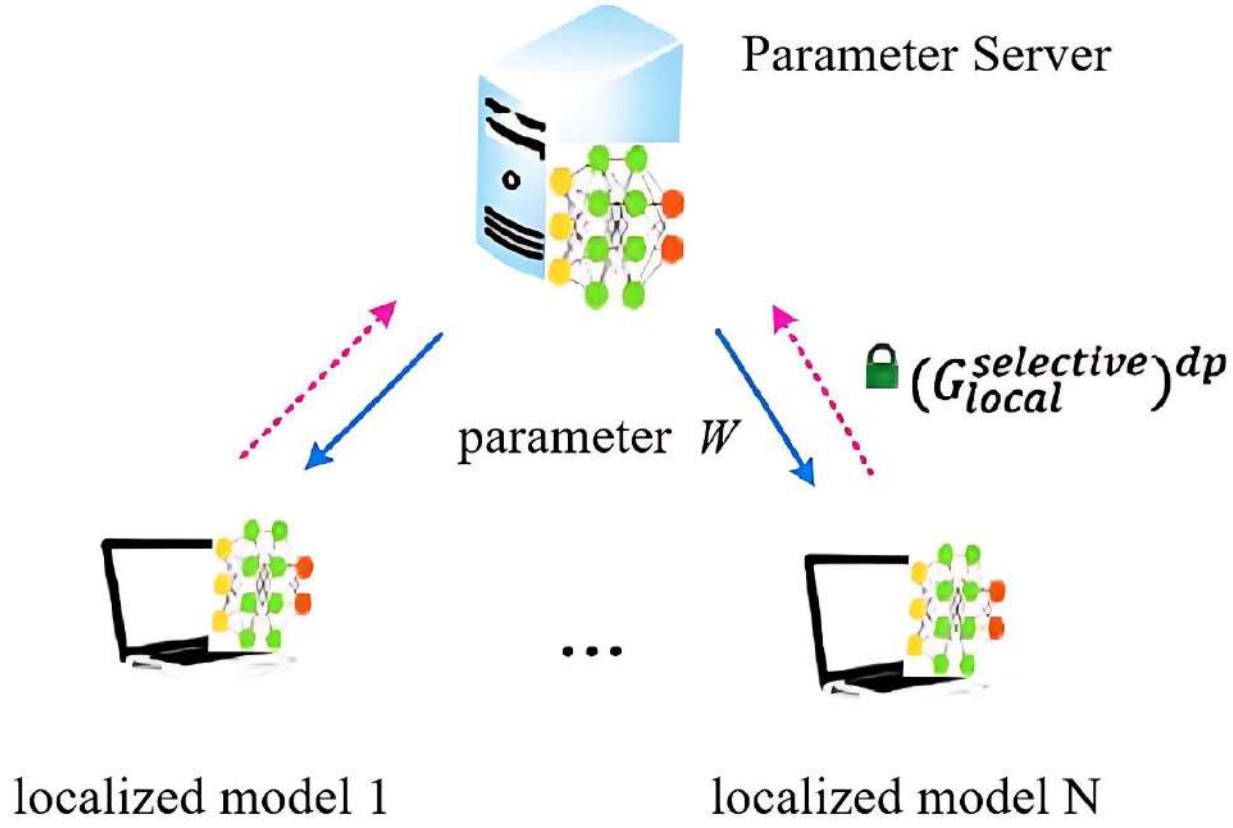

Figure 4.
Framework for DSSGD.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 564-594, 2025
DOI: 10.55214/25768484.v9i1.4184

In Shokri and Shmatikov [32] the magnitude of injected noise and the privacy budget are accumulated in proportion to the number of training epochs and the amount of shared parameters. Consequently, this approach may result in the unnecessary consumption of a substantial privacy budget, as training iterations and the number of shared parameters among parties which are typically large. To address this issue and track privacy loss during the training process, Abadi, et al. [78] proposed a Moments Accountant (MA) mechanism based on the composition theorem. This mechanism enables automatic tracking and analysis of privacy loss, providing a tighter estimate of overall privacy loss than advanced composition theorems.

The core idea involves the use of a differentially private stochastic gradient descent algorithm (Differentially Private SGD), where noise is added to the "gradient" parameters at each training step. The MA mechanism is then used to perform fine-grained, automated tracking of cumulative privacy loss during training, thereby allowing each participant to manage particularly sensitive gradient parameters and ensuring that parameter sharing does not result in significant privacy leakage. This approach allows deep models with millions of parameters to be trained under controllable privacy costs, even in scenarios involving strong adversaries who might control some or all of the remaining training data. Experiments on the MNIST dataset achieved 97% training accuracy.

However, the method proposed in Abadi, et al. [78] still relies on the number of training epochs. When privacy budgets are limited, only a small number of iterations can be used for model training. This limitation may adversely impact the utility of the model when a high number of training iterations is necessary to ensure accuracy. Additionally, another drawback of the existing approach is that the same amount of noise is injected into all parameters. This uniform noise injection may not be ideal in practical scenarios, as different features and parameters often have varying impacts on the model's output.

To address these limitations, Phan, et al. [79] proposed an adaptive Laplace mechanism (AdLM) for differential privacy protection in deep neural networks, based on the layer-wise relevance propagation (LRP) algorithm [80]. The LRP framework is illustrated in Figure 5.

The implementation of AdLM involves the following steps:

- Relevance Evaluation: Using the principles of the LRP algorithm, affine transformation, and back propagation theory, the relevance between each input feature x_{ij} and the model output $F_{xi}(\theta)$ is evaluated, as shown in Equation (5).
- Noise Addition Based on Relevance: The average relevance Rj of each feature over the dataset D is computed using a pre-trained neural network, and Laplace noise is added, as shown in Equation (6).
- Adaptive Noise Injection: Based on the varying contributions of each feature x_{ij} to the output, noise is adaptively injected into the features. Features less relevant to the model's output are assigned more Laplace noise, as shown in Equation (7). Unlike earlier methods, AdLM ensures that the noise injected and the corresponding privacy budget consumption at each training step do not accumulate. Therefore, the privacy budget consumption is entirely independent of the number of training iterations.

$$\mathcal{F}_{x_i}(\theta) = \sum_{m \in h_k} R_m^{(k)}(x_i) = \cdots = \sum_{x_i \in x_i} R_{x_i}(x_i) \quad (5)$$

$$\overline{R_j} \triangleq \frac{1}{|D|} \sum_{x_j \in D} R_{x_j}(x_i) + \mathrm{Lap}(\Delta R/\varepsilon_1) \quad (6)$$

$$\widehat{x_{ij}} \stackrel{\Delta}{=} x_{ij} + \frac{1}{|L|} \mathrm{Lap}\left(\Delta h_0/\varepsilon_j\right) \quad (7)$$

Here, the noise coefficient $\epsilon j = \beta j \times \epsilon$, where ϵ represents the total noise injected at the current step, and βj denotes the contribution coefficient of the j-th feature to the output of the neuron.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 564-594, 2025
DOI: 10.55214/25768484.v9i1.4184

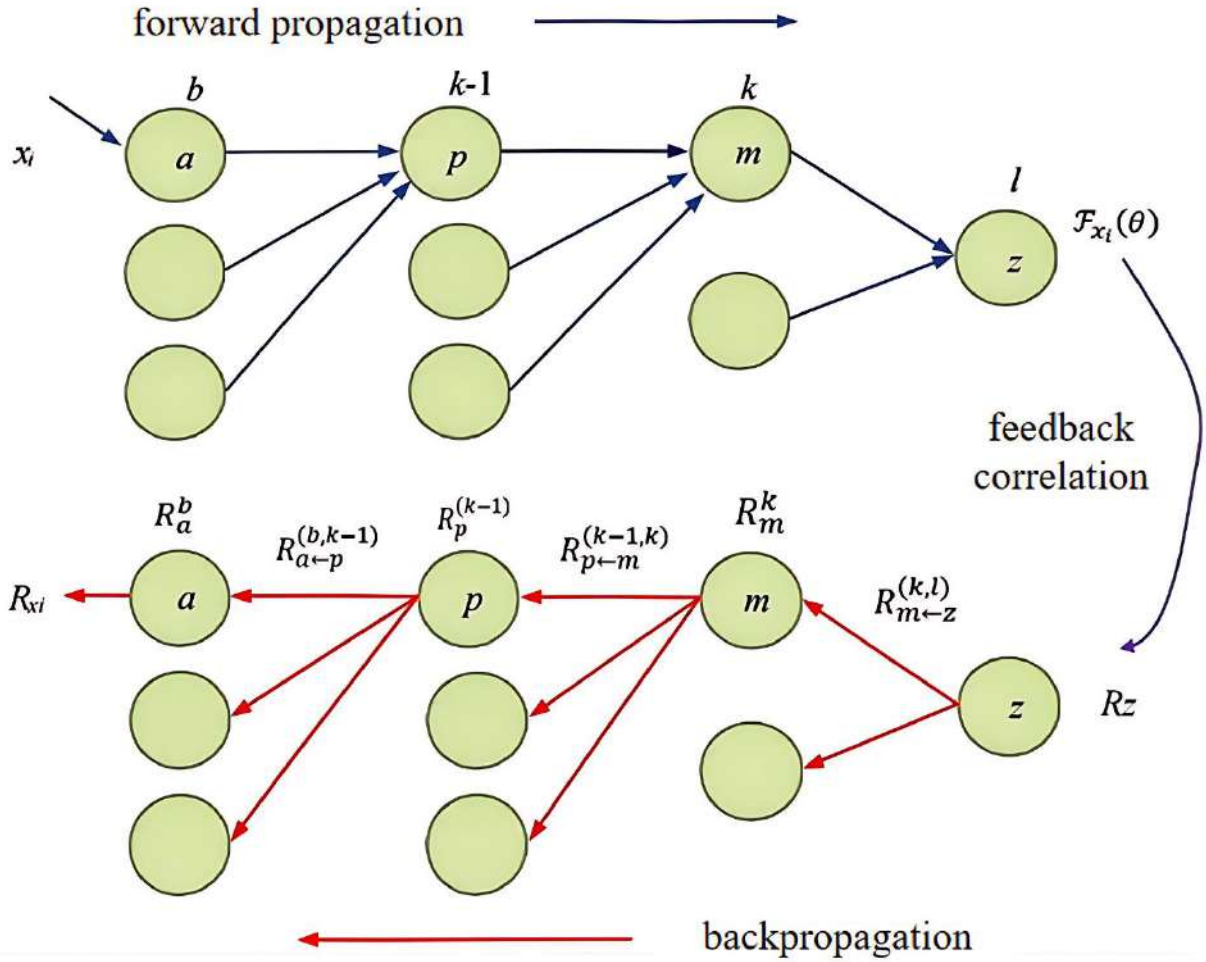

Figure 5.
Framework for LRP.

Privacy Protection Schemes Based on Objective Perturbation: Objective perturbation, also known as function perturbation, involves adding Laplace noise to the coefficients of the objective function or its expanded representation in a machine learning model, followed by minimizing this perturbed objective function. Unlike parameter perturbation, the privacy loss in objective perturbation is determined by the objective function itself and is independent of the number of training iterations. Studies [81] have demonstrated that objective perturbation, under theoretical guarantees, is more effective than output perturbation methods. However, objective perturbation requires the objective function to be continuously differentiable and convex, which limits its application to non-convex models such as neural networks.

An alternative approach is to add Laplace noise to the coefficients of the expanded representation of the objective function. To inject noise into the coefficients, the objective function must be expressed as a polynomial of the weights. For non-polynomial objective functions, approximation techniques such as Taylor expansion or Chebyshev expansion are employed to convert the function into a polynomial form, after which noise is added to the coefficients. However, this method is constrained by its reliance on specific objective functions, making it difficult to generalize to more versatile models.

Chaudhuri and Monteleoni [40] leveraging sensitivity-based methods [82] designed a privacy-preserving logistic regression algorithm. This approach involves defining the sensitivity of the function

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 564-594, 2025
DOI: 10.55214/25768484.v9i1.4184

to be learned and then perturbing the classifier with noise proportional to the sensitivity. While the ϵ\epsilonϵ-differential privacy model in this method limits adversaries from obtaining specific private information, it can be challenging for certain machine learning functions. To address this, the authors proposed an alternative privacy-preserving logistic regression method based on a perturbed objective function, which does not rely on function sensitivity and ensures privacy within the model. Experiments demonstrated that this method achieves superior learning performance.

To counter potential model inversion attacks in deep learning, Phan, et al. [83] focused on auto-encoders, a fundamental deep learning component, and proposed the Deep Private Autoencoder (dPAs) framework. This approach applies ϵ\epsilonϵ-differential privacy to perturb the cross-entropy error function of deep auto-encoders, introducing noise during the data reconstruction process to protect training data privacy. For objective functions with infinite polynomial terms, Taylor expansion is used for approximation.

The applicability of existing DP algorithms to deep learning has attracted significant attention. For instance, the dPAs framework [83] was specifically designed for certain deep learning models. To expand its scope, Phan, et al. [49] proposed the Private Convolutional Deep Belief Network (pCDBN). The Convolutional Deep Belief Network (CDBN) is a representative energy-based deep learning model, with a more complex structure than auto-encoders. The pCDBN, essentially a differentially private CDBN, employs Chebyshev expansion to approximate the nonlinear objective function as a polynomial and inject noise into the polynomial coefficients. Each hidden layer satisfies ϵ\epsilonϵ-differential privacy during training. The pCDBN framework ensures that the privacy budget is independent of the number of training epochs, making it suitable for large datasets and significantly advancing the application of privacy-preserving techniques in deep learning.

Jayaraman, et al. [84] proposed a method for applying differential privacy perturbation to distributed learning outputs. Multiple parties collaboratively train a machine learning model using secure multi-party computation protocols, with Laplace noise added to the global model output for perturbation. The authors demonstrated that introducing noise into the aggregated model in secure multi-party computation scenarios requires less noise than other perturbation methods and prevents adversaries from conducting inference attacks on the final model. Experiments on the KDD Cup 98 dataset showed that this method achieves accuracy comparable to non-private methods.

Papernot, et al. [56] inspired by semi-supervised knowledge transfer, proposed the Private Aggregation of Teacher Ensembles (PATE) model to address privacy leakage in training data. PATE divides sensitive data into NNN disjoint subsets, training a teacher model on each subset. For public data to be labelled, differential privacy noise is added to the collective voting results of the teacher models, with the class receiving the most votes used as the prediction result. A student model is then trained using the labeled dataset and deployed for predictions, preventing adversaries from reverse-engineering sensitive data. However, the privacy loss of PATE is proportional to the volume of labeled public data, potentially leading to significant privacy costs, restricting its applicability to simple classification tasks.

Papernot, et al. [85] later extended PATE to large-scale environments, enabling its use in image classification tasks. The improved PATE outperformed the original model across multiple metrics by introducing a new noise aggregation mechanism, Rényi Differential Privacy (RDP) [86]. RDP offers stricter differential privacy guarantees at lower privacy costs compared to traditional methods. A key limitation of the PATE framework is its reliance on the availability of unlabelled, non-sensitive public data with statistical properties similar to the teacher model's training data, which is often unrealistic in domains such as healthcare.

Differential privacy, in contrast to encryption techniques, achieves privacy protection by randomizing data with noise perturbation, imposing minimal additional computational overhead compared to traditional non-private algorithms [87]. However, it can reduce model utility, leading to a decline in prediction accuracy. The strictest differential privacy mechanisms can protect machine learning models from membership inference or inversion attacks, even in cases where adversaries know all but one record in the dataset. However, such stringent measures can render models unusable [13].

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 564-594, 2025
DOI: 10.55214/25768484.v9i1.4184

A practical solution is to relax privacy protection requirements, allowing algorithms to satisfy looser differential privacy constraints, though this increases the risk of privacy leakage [88]. Localized differential privacy techniques mitigate the risk of data theft during data collection by perturbing data or intermediate results before uploading to servers. This prevents direct access to raw user data while enabling statistical analysis. Deep learning models, characterized by non-convex objective functions, numerous parameters, and complex structures, require extensive access to sensitive training data and multiple training iterations to converge to an optimal, often local, solution. Ensuring differential privacy in every parameter update imposes a significant global privacy cost, creating a challenging trade-off between privacy and model utility. Differential privacy-enabled generative adversarial networks produce synthetic data with limited privacy guarantees, as these data often closely resemble real samples and are susceptible to statistical inference attacks due to their preserved statistical properties.

3.2. Privacy Protection Mechanisms for Machine Learning Based on Homomorphic Encryption

The interplay between cryptography and machine learning has been studied extensively, with these two fields often considered opposites. Cryptography aims to prevent access to information, whereas machine learning seeks to extract insights from data [89]. In the field of machine learning, ensuring user data confidentiality can be achieved using traditional cryptographic methods. However, these methods require encryption and decryption phases, making them impractical in real-world scenarios due to their high computational complexity. Recent advancements in cryptography have introduced techniques that allow arbitrary operations on encrypted data without decryption. Full homomorphic encryption (FHE) is one such method. This section first provides an overview of homomorphic encryption (HE) techniques, followed by a review of research progress in privacy protection for machine learning based on HE.

Homomorphic encryption (HE) is a cryptographic technique that allows operations to be performed directly on ciphertext, yielding results that remain encrypted. When decrypted, these results match those obtained from performing the same operations on plaintext. Homomorphic encryption schemes satisfy the following equation (Equation 8):

$$\mathrm{Dec}\left(k_s, \mathrm{Enc}(k_p, m_1) \diamond \mathrm{Enc}(k_p, m_2)\right) = m_1 \circ m_2 \quad (8)$$

Here, m1 and m2 represent plaintexts, ks and kp are the secret and public keys, respectively, Enc () denotes encryption, Dec ()\denotes decryption, ⋄ and ∘ represents the operation performed. Depending on the stage of development and the types and frequencies of operations supported on ciphertext, HE is categorised into partially homomorphic encryption (PHE), somewhat homomorphic encryption (SHE), and fully homomorphic encryption (FHE) [90].

- Partially Homomorphic Encryption (PHE): PHE was the earliest homomorphic encryption scheme and supports only addition or multiplication operations, though these operations can be performed an unlimited number of times. PHE schemes are further divided into: (1) Additive Homomorphic Encryption (AHE): For example, the Paillier scheme; (2) Multiplicative Homomorphic Encryption (MHE): For example, the ElGamal scheme.
- Somewhat Homomorphic Encryption (SHE): SHE supports a limited number of additions and multiplications on ciphertext. Although SHE is less robust than FHE, it incurs lower computational costs and is easier to implement. Leveled fully homomorphic encryption (leveled-FHE), also referred to as bounded-depth homomorphic encryption, is a subset of SHE schemes [91]. "Bounded depth" refers to the ability to handle only a finite number of circuit depths, making leveled-FHE unsuitable for training deep neural networks. However, leveled-FHE supports single instruction multiple data (SIMD) batch processing, resulting in relatively high performance.
- Fully Homomorphic Encryption (FHE): FHE, introduced by Gentry [92] based on ideal lattice theory, supports arbitrary algorithms on ciphertext with an unlimited number of operations. While FHE is secure and reliable, the bootstrapping process—a key component—is computationally expensive. This significant computational overhead makes FHE impractical for

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 564-594, 2025
DOI: 10.55214/25768484.v9i1.4184

real-world applications, particularly in big data environments. In recent years, several improved FHE schemes [93-96] have been proposed, focusing on noise reduction and efficiency enhancement.

Table 4 provides a comparative analysis of privacy protection mechanisms for machine learning based on homomorphic encryption. It evaluates these mechanisms from perspectives such as key technologies, support for deep models, batch processing capabilities, support for non-linear operations, and classification accuracy.

Homomorphic encryption represents a promising direction for enabling privacy-preserving machine learning by allowing encrypted data to be processed directly. However, challenges such as high computational costs and limited practicality in large-scale data scenarios remain focal points of ongoing research and development.

Table 4.
Comparison of machine learning privacy protection schemes based on cryptography.

Category	Key technology	Support for deep models	Batch processing	Support for Non-linear operations	Dataset	Accuracy (%)
Without polynomial approximation	HE	No	No	Yes	N/A	N/A
	HE	No	N/A	Yes	UCI Datasets	99.8
	AHE	N/A	N/A	N/A	WBC	98.24
	AHE	No	No	N/A	N/A	N/A
	FHE	Yes	Yes	N/A	N/A	N/A
	FHE, BNNs	Yes	Yes	Yes	MNIST	99.04
With polynomial approximation	Leveled-FHE	No	Yes	No	N/A	N/A
	FHE	Yes	N/A	No	STL-10	85.5
	HE	Yes	Yes	No	MNIST	99.10
	AHE	No	Yes	N/A	MNIST	99
	Leveled-FHE	No	Yes	No	MNIST	98.95
	FHE	Yes	Yes	No	MNIST	99.30
	Leveled-FHE	No	Yes	No	MNIST	99.52

Homomorphic Encryption Privacy Protection Schemes Without Polynomial Approximation: Homomorphic encryption (HE) schemes are secure and reliable but support only polynomial operations, such as addition and multiplication, making them incompatible with non-linear operations commonly used in machine learning, such as the sigmoid and ReLU activation functions in neural networks. One solution involves relying on data owners to perform non-linear operations. For instance, Barni, et al. [53] proposed a privacy-preserving method for neural networks in which data owners encrypt data using HE and send it to the cloud platform. The platform computes the inner product between the data and the first layer's weights and sends the result back to the data owner. The data owner decrypts the result, applies a non-linear transformation, re-encrypts it, and sends it back to the cloud. This process is repeated for all layers. However, this requires the data owner to remain online and share intermediate results, potentially exposing most of the neural network's weight information to the data owner.

To address this limitation, Orlandi, et al. [46] proposed an alternative privacy-preserving method. HE is used to encrypt data, ensuring confidentiality. An interactive protocol between the data owner and the model owner resolves the issue of non-linear activation functions. In this scheme, the model sends encrypted inputs to the data owner for non-linear transformation. The data owner decrypts the input, applies the transformation, re-encrypts the result, and sends it back. Unfortunately, this interaction incurs significant latency, increases complexity on the data owner's side, and leaks some model information. To mitigate this, Orlandi et al. introduced security measures, such as random execution orders.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 564-594, 2025
DOI: 10.55214/25768484.v9i1.4184

To prevent privacy leakage during classification, Rahulamathavan, et al. [9] developed a scheme using the Paillier encryption system to transform SVM decision functions into ciphertext. Classification samples are also encrypted, and all computations occur on ciphertext, ensuring only the tester holding the private key can decrypt and access the classification results. Prasad, et al. [43] employed additive homomorphic encryption (AHE) with the Naïve Bayes algorithm to address privacy in distributed communication environments for both continuous and discrete data. Similarly, Aslett, et al. [44] utilised Bayesian classifiers, random forests, and their variants to train machine learning models on FHE-encrypted data. While effective for specific tasks, such as text classification, their performance in image recognition was inferior to that of neural networks.

Binary Neural Networks with Homomorphic Encryption: To facilitate deep learning on devices with limited memory and computational resources, such as mobile phones, binary neural networks (BNNs) have gained popularity [97, 98]. By binarising weights and activations (taking values of +1 or -1), BNNs reduce memory usage significantly, requiring only 1 bit per value instead of 32-bit floating-point representation. Combining BNNs with HE enables efficient and accurate predictions on encrypted data. Chillotti, et al. [99] proposed a bootstrapped FHE scheme optimised with the TFHE library, reducing bootstrapping time to under 0.1 seconds. This scheme supports operations on binary data, facilitating all BNN operations.

Bourse, et al. [100] introduced the FHE-DiNN model, which performs encrypted predictions using BNNs. While the model achieved moderate accuracy on MNIST, the encryption scheme parameters were tightly coupled with the model structure, requiring users to re-encrypt data whenever the model was updated. Sanyal, et al. [25] proposed the TAPAS system, which integrates FHE-encrypted data with binary and sparse techniques to optimise neural network design. The system achieved accelerated parallel computations and allowed service providers to update models dynamically, attaining 99.04% accuracy on MNIST. Both FHE-DiNN and TAPAS leverage the BNN concept and outperform leveled-FHE batch prediction methods in prediction speed.

Homomorphic Encryption for Secure Outsourced Computation: For outsourced computation environments, some schemes fail to guarantee the privacy of machine learning models [68, 69] or intermediate computation results [101]. To address this, Li, et al. [102] proposed a privacy-preserving convolutional neural network (CNN) prediction scheme. This approach combines HE, secret sharing, and garbled circuits to store prediction data and initial model parameters securely across two non-colluding servers. These servers collaboratively perform model predictions while remaining unaware of user data and model parameters. Non-linear activation functions, such as ReLU, are handled using garbled circuits instead of polynomial approximations, achieving plaintext-equivalent accuracy. Techniques such as data packing, single instruction multiple data (SIMD), and asynchronous computation reduce computational and communication overheads, improving speed. Liu, et al. [103] proposed a similar scheme, ensuring the privacy of outsourced data, intermediate queries, and results, alongside the model itself.

Homomorphic Encryption in Neural Network Training: Zhang, et al. [62] employed the BGV FHE scheme to train deep computational models directly on ciphertext. They approximated non-polynomial functions, such as activation functions, using Taylor expansions, enabling efficient and secure computation of higher-order backpropagation algorithms. To limit excessive multiplication depth, updated weights were decrypted and re-encrypted after each iteration, increasing communication complexity. Hesamifard, et al. [61] used Chebyshev polynomial approximations for activation functions during neural network training. Substituting ReLU with polynomial approximations yielded 99.10% prediction accuracy, while sigmoid approximations achieved 99.00%.

To mitigate privacy risks in Shokri and Shmatikov [32] where even minimal gradient information could reveal user data through reverse model attacks, Trieu, et al. [60] developed a homomorphic encryption-based privacy-preserving deep learning (PPDL) system. Participants encrypt gradient parameters using AHE and send them to a central server, safeguarding against untrusted servers. The framework, illustrated in Figure 8, uses AHE for weight updates, significantly accelerating DNN

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 564-594, 2025
DOI: 10.55214/25768484.v9i1.4184

training via parallel gradient computation across processing units. However, the enhanced privacy comes with increased communication costs.

Homomorphic Encryption in Neural Network Predictions: Gilad-Bachrach, et al. [51] proposed the CryptoNets model, using leveled-FHE (YASHE [104]) for approximate neural network predictions. Trained neural network models on plaintext were approximated with low-degree polynomials, enabling encrypted predictions. Although CryptoNets achieved 98.95% accuracy on MNIST, its reliance on leveled-FHE increased computational complexity, particularly for deep networks. Chabanne, et al. [54] extended this approach, combining polynomial approximations of ReLU activation functions with batch normalisation for deeper networks. While effective, these methods required clients to generate encryption parameters based on the model structure, risking model privacy leakage.

Hesamifard, et al. [47] introduced CryptoDL, which classifies FHE-encrypted data using pre-trained plaintext models. Low-degree polynomial approximations of activation functions improved classification efficiency, aided by SIMD batch processing.

Homomorphic encryption is a promising end-to-end encryption system that fundamentally addresses trust issues in data models. It empowers users to retain control over their data while benefiting from remote server computations. In centralised machine learning, users upload encrypted training data to servers for model training without exposing raw data. Similarly, in federated learning, encrypted parameters or gradients uploaded to a central server facilitate model training while protecting user and model privacy.

While any computation can be expressed as a binary polynomial, HE supports only integer operations, requiring non-linear functions, such as activation functions, to be approximated with polynomials. These approximations degrade accuracy and efficiency. HE entails significant computational and communication overheads, posing challenges for current resources. However, leveled-FHE, supporting SIMD batch processing, offers higher performance with optimised parameters and minimal operations. Despite these advancements, training deep neural networks on ciphertext remains a significant open problem due to the computational cost of non-linear functions like sigmoid or softmax.

3.3. Privacy-Preserving Mechanisms for Machine Learning Based on Secure Multi-Party Computation (SMC)

This type of scheme is essentially an encrypted distributed machine learning technique. Participants collaboratively construct a unified machine learning model over the entire dataset by exchanging necessary information without disclosing their private data. Vaidya and Clifton [57] proposed an SMC-based k-means clustering algorithm for arbitrarily partitioned data. The method enables parties to collaboratively perform k-means computations across the entire dataset without revealing their respective data. Similarly, Bansal, et al. [105] introduced a neural network learning algorithm based on Homomorphic Encryption (HE) for arbitrarily partitioned training datasets. This approach ensures that no data privacy is leaked, including intermediate results, except for the final trained weights, which are known to both parties.

Samet and Miri [42] developed an extreme learning machine tailored for horizontally or vertically partitioned training data. However, due to direct participation by data holders in operations not supported by partial homomorphic encryption, this method risks leaking sensitive information about the learning model. Mehnaz, et al. [106] proposed a generic framework for privacy-preserving model training on partitioned datasets. This framework incorporates two secure gradient descent algorithms—one for horizontally partitioned data and the other for vertically partitioned data. The solution is resilient to collusion attacks and suitable for large-scale multi-party computation scenarios and various machine learning algorithms.

Enhancing the computational efficiency of SMC algorithms is a focal point of contemporary research in machine learning. Li, et al. [41] proposed an outsourced computation solution based on an improved C4.5 decision tree. The authors employed OPPWAP and OSSIP protocols to implement general-purpose SMC computations, outsourcing computational tasks to a server using cryptographic algorithms. The distributed C4.5 decision tree for horizontally partitioned data was reduced to a

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 564-594, 2025
DOI: 10.55214/25768484.v9i1.4184

weighted averaging problem, while for vertically partitioned data, it was reduced to a secure intersection problem, thereby lowering the computational complexity on the user side to sublinear levels.

Abbasi, et al. [107] introduced a Secure Clustered Multi-Party Computation (SCMC) method that allows a certain degree of privacy leakage within clusters, striking a balance between efficiency and privacy preservation. Asharov, et al. [108] employed extended Oblivious Transfer (OT) protocols within various SMC models, reducing communication and computational complexity. Experimental results demonstrated that the improved OT algorithms significantly enhanced the efficiency of SMC systems.

Gheid and Challal [58] addressed the privacy leakage associated with directly running k-means clustering algorithms on large datasets by proposing an improved secure multi-party summation protocol. This protocol is straightforward and mitigates performance degradation caused by cryptographic solutions. Dani, et al. [109] leveraged the quorum concept to design synchronous and asynchronous SMC protocols. These protocols address the issue of linear growth in communication and computational costs with an increasing number of participants, making them suitable for large-scale distributed systems. The approach achieves security while reducing communication and computational complexity from linear to sublinear levels.

Bogdanov, et al. [110] utilised the advantages of the Sharemind model to enable secure computations for large datasets, overcoming the limitations of general SMC models in handling large-scale data. However, Sharemind supports only three-party computations and does not extend to scenarios involving more participants.

SMC Schemes Based on the Two-Party Computation (2PC) Architecture: SMC schemes based on the 2PC architecture represent another prominent category of multi-party computation privacy-preserving solutions. These schemes integrate several foundational cryptographic protocols for secure multi-party computation. Classical two-party computation approaches include combinations such as HE+GC [39], HE+GC+SS+OT [111], GC+OT [67], HE+GC+SS [69] and GC+SS+OT [63]. In these schemes, one party serves as the data provider, while the other acts as the computational server.

Table 5 compares privacy-preserving SMC schemes based on the 2PC architecture across various dimensions, including key technologies, support for non-linear operations, batch processing capabilities, runtime, communication overhead, and accuracy.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 564-594, 2025
DOI: 10.55214/25768484.v9i1.4184

Table 5.
Comparison of SMC privacy-preserving schemes based on 2PC.

Key Technology	Supports non-linear operations	Supports batch processing	Dataset	Offline runtime (s)	Online runtime (s)	Total runtime (s)	Offline communication (MB)	Online communication (MB)	Total communication (MB)	Accuracy (%)
HE, GC, SS, OT	No	No	MNIST	4.7	0.18	4.88	N/A	N/A	N/A	93.4
GC, OT	Yes	No	MNIST	N/A	9.67	9.67	N/A	791	791	98.95
AHE, GC, SS	Yes	No	MNIST	3.58	5.74	9.32	20.9	636.6	657.5	99
GC, SS, OT	Yes	N/A	MNIST	N/A	N/A	5.1	N/A	N/A	501	99.2
GC, GMW, A-SS	Yes	No	MNIST	1.25	0.99	2.24	5.4	5.1	10.5	99
AHE, GC	Yes	Yes	MNIST	0.48	0.33	0.81	47.5	22.5	70.0	N/A
A-SS	Yes	No	MNIST	0.09	0.21	0.3	1.57	0.99	2.56	99.14
A-SS	Yes	No	TIMIT	N/A	N/A	0.39518	N/A	N/A	2.435	N/A

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 564-594, 2025
DOI: 10.55214/25768484.v9i1.4184

Nikolaenko, et al. [39] proposed a horizontally partitioned data privacy-preserving linear regression algorithm based on leveled Fully Homomorphic Encryption (leveled-FHE) and Garbled Circuits (GC). Experiments on datasets with millions of samples demonstrated that this method significantly outperforms privacy-preserving schemes based solely on leveled-FHE or GC. The approach supports scalability in terms of the number of users and features while maintaining the accuracy of results.

Mohassel, et al. [111] introduced SecureML, a dual-server machine learning framework designed with techniques such as Secure Multi-Party Computation (SMC), Secret Sharing (SS), and multiplication triplets. The scheme encrypts data with leveled-FHE and distributes it to two non-colluding servers, enabling secure two-party computations to train models such as neural networks. While the framework primarily supports model training, it also facilitates privacy-preserving predictions. During training, polynomial approximations are used for non-linear activation functions, with precomputation reducing computational costs in the online prediction phase. The scheme achieves performance improvements of 1,100 to 1,300 times over the protocol in Nikolaenko, et al. [39] and scales to datasets with millions of samples. However, it leaks some model information through prediction outputs.

Chandran, et al. [63] developed a secure two-party computation framework, EzPC, which generates efficient two-party computation protocols from high-level, user-friendly programs. EzPC combines arithmetic sharing with garbled circuits, ensuring that servers cannot access the client's input or output information, while clients are also denied access to the server's model details. The protocols generated by the EzPC framework are 19 times faster than existing protocols supporting secure prediction and matrix decomposition.

Henecka, et al. [70] proposed TASTY, an automated tool that leverages Homomorphic Encryption (HE) and GC to generate secure two-party computation protocols for specific applications, including Private Set Intersection (PSI) and privacy-preserving face recognition. TASTY provides automation in protocol description, generation, execution, benchmarking, and comparison.

Addressing the decline in model accuracy caused by approximating non-linear activation functions during training in frameworks like CryptoNets [51] and SecureML [111], Rouhani, et al. [67] proposed DeepSecure, a framework for oblivious neural network prediction. This framework, based on GC, supports any non-linear activation function without modifying the neural network training process, thus preserving model accuracy. DeepSecure eliminates the risk of collusion attacks present in dual-server systems like SecureML [111]. To reduce the overhead of GC protocols, the framework introduces a preprocessing phase.

Liu, et al. [15] proposed MiniONN, a two-party computation framework based on oblivious neural networks (ONN). MiniONN incorporates HE methods during the offline precomputation phase and lightweight cryptographic primitives like secret sharing during the online prediction phase, ensuring the privacy of both models and data. By employing the authentic sigmoid activation function for training, the framework avoids modifications to the neural network training process.

Riazi, et al. [65] introduced Chameleon, a hybrid secure computation framework aimed at reducing the overhead of GC protocols. Chameleon employs Additive Secret Sharing (A-SS) for linear operations and either Goldreich-Micali-Wigderson (GMW) or GC protocols for non-linear operations. Similar to SecureML [111], Chameleon relies on a semi-honest third party (STP) to generate correlated randomness during the offline phase. By offloading most cryptographic operations to the offline stage, the framework significantly reduces computational and communication overheads, thereby improving classification efficiency.

Juvekar, et al. [68] proposed GAZELLE, a secure neural network inference framework based on Additive Homomorphic Encryption (AHE) and GC. This framework allows clients to obtain encrypted classification results without revealing their inputs to the server, ensuring the privacy of neural networks. GAZELLE performs linear operations using AHE and non-linear operations using GC. By employing Single Instruction Multiple Data (SIMD) operations and avoiding ciphertext-ciphertext multiplication, it mitigates noise growth. Compared to purely homomorphic approaches like CryptoNets

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 564-594, 2025
DOI: 10.55214/25768484.v9i1.4184

[51], GAZELLE reduces latency by three orders of magnitude and bandwidth by two orders of magnitude.

Huang, et al. [52] addressed the challenges of frameworks like Riazi, et al. [65] and Juvekar, et al. [68] which rely on computationally intensive cryptographic primitives, making it difficult to leverage the efficient parallel data structures of Convolutional Neural Networks (CNNs). These frameworks are also less suitable for deployment on resource-constrained mobile sensors. To overcome these limitations, Huang et al. introduced a lightweight privacy-preserving framework, LPP-CNN, designed for CNN feature extraction in edge computing-based mobile sensors. The system architecture, depicted in Figure 6, employs A-SS and multiplication triplets to design a series of efficient secure interactive subprotocols. Two edge servers and a trusted third party collaboratively execute CNN feature extraction, with the trusted third party generating random values during the offline phase. Since no approximations are required for the CNN structure, the framework preserves the accuracy of the CNN model while significantly reducing computational and communication costs by avoiding reliance on computationally intensive cryptographic primitives.

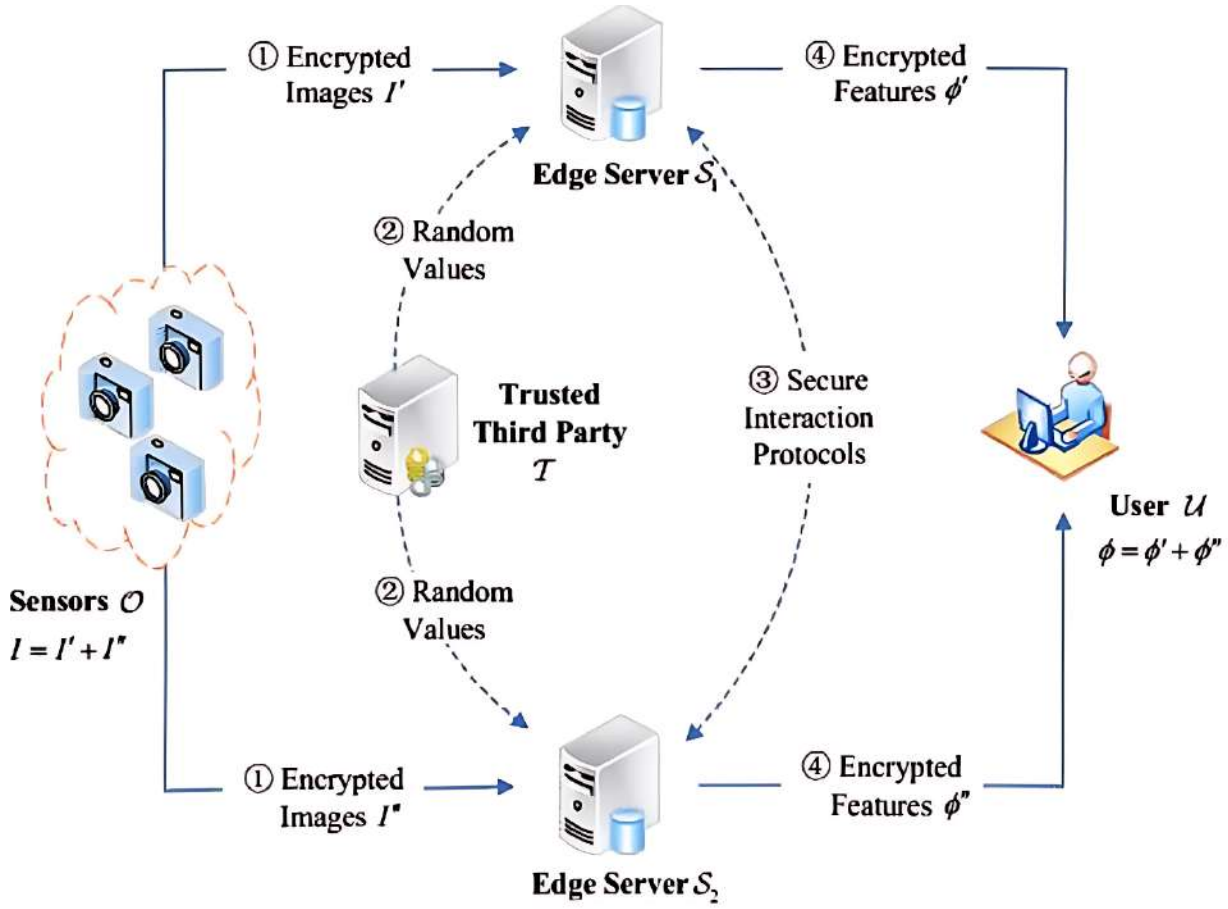

Figure 6.
System architecture for LPP-CNN.

To address privacy concerns in edge computing, Ma, et al. [66] proposed a lightweight privacy-preserving framework, POR. This framework leverages Additive Secret Sharing (A-SS) and edge computing technologies to perform collaborative AdaBoost-based facial recognition classification tasks using two servers. A series of interactive protocols was designed for different training stages of

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 564-594, 2025
DOI: 10.55214/25768484.v9i1.4184

AdaBoost. Experimental results demonstrated that POR reduces computational error by approximately 58% compared to existing differential privacy-based frameworks.

Another lightweight privacy-preserving framework, OPSR, was introduced in Ma, et al. [55] for privacy-preserving voice recognition in intelligent Internet of Things (IoT) devices. OPSR employs Long Short-Term Memory (LSTM) neural networks, edge computing, and A-SS technologies to facilitate lightweight outsourced computation tasks. Compared to frameworks based on Homomorphic Encryption (HE) and Garbled Circuits (GC), OPSR significantly reduces computational and communication overheads.

For energy management strategy formulation in smart grids, [59] proposed a lightweight privacy-preserving reinforcement learning framework, LiPSG. This framework is based on A-SS and edge computing technologies. In LiPSG, the power data of each supply region is securely outsourced to a third-party dual server for Q-learning model computation before being sent to the control centre. Throughout the Q-learning process, the data remains in a randomly shared format, preventing adversaries from misusing user data.

Secure Multi-Party Computation (SMC) protocols enable multiple participants to perform aggregated computations on data without sharing their inputs. These protocols employ cryptographic techniques such as Homomorphic Encryption, Secret Sharing, and Oblivious Transfer. SMC protocols are particularly suitable for efficient parallel distributed machine learning and have been shown to scale to learning tasks involving billions of records in certain environments [112].

However, unlike differential privacy-based approaches that protect models from inference attacks, SMC-based methods primarily safeguard the privacy of training data during the learning process [84]. The key to constructing SMC protocols that support multi-party collaborative training of machine learning models lies in the following: (1) Selecting Appropriate Cryptographic Tools Based on Protocol Characteristics: Homomorphic Encryption is effective for linear operations, particularly matrix-vector multiplication, as it has low overall communication complexity [68]. It is well-suited for specific additive and multiplicative operations on large numerical values; (2) Designing Efficient Alternatives for Non-Linear Functions in Machine Learning Models: Existing fixed-point or floating-point multiplication techniques require bit-level operations, which are most efficiently implemented using Boolean circuits [111]. In theory, any function representable as a Boolean circuit can be securely computed using garbled circuit protocols [65]. Consequently, garbled circuits are particularly suitable for approximating non-linear functions in Deep Neural Networks (DNNs); (3) Extending SMC Protocols Beyond Two-Party Scenarios: Privacy-preserving schemes based on two-party computation (2PC) architectures, such as HE+GC+SS+OT, provide robust privacy protection. However, they are predominantly designed for two-party scenarios, and extending these protocols to multi-party settings results in significant communication overhead. Moreover, due to their reliance on complex technologies, these protocols are relatively slow and typically unsuitable for large-scale datasets [106].

Privacy-preserving technologies such as Differential Privacy, Homomorphic Encryption, and Secure Multi-Party Computation possess distinct technical characteristics, advantages, and limitations, making them suitable for different application scenarios. Table 6 provides a comparative analysis of these technologies in the context of machine learning.

Table 6.
Comparison of different privacy-preserving technologies.

Technology	Technical characteristics	Advantages	Disadvantages	Privacy-preserving scenarios
Differential privacy	Noise-perturbed data	High privacy and efficiency	Degrades classifier performance; low usability	Environments with limited computational resources
Homomorphic encryption	Computation on ciphertext	High privacy	High computational and storage overhead; low efficiency and usability	High-computation environments with stringent privacy requirements
Secure multi-party computation	Collaborative computation without exposing privacy	Good privacy and usability	High communication overhead; low efficiency	Distributed collaborative learning environments

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 564-594, 2025
DOI: 10.55214/25768484.v9i1.4184

When applying privacy-preserving technologies in practice, it is essential to consider factors such as the hardware performance of user devices, transmission costs, and time constraints. For instance, organisations with large volumes of sensitive data, such as hospitals or banks, may require the use of Homomorphic Encryption (HE) to ensure the security of the model. Conversely, when individual users with limited computational capabilities are involved, Differential Privacy (DP) may be employed to maintain model efficiency. In distributed environments where multiple parties collaboratively train a machine learning model, Secure Multi-Party Computation (SMC) may be utilised to safeguard the privacy of all participants. Increasingly, research is focusing on combining methods such as SMC, HE, and DP to achieve a balanced trade-off between data privacy and utility.

4. Conclusion

Machine learning has become a core technology in big data, the Internet of Things (IoT), cloud computing, and artificial intelligence. The privacy threats associated with machine learning, as well as corresponding defence mechanisms, have attracted increasing attention from academia and industry. Despite this progress, research on privacy preservation in machine learning is still in its infancy, and many critical issues remain unresolved. Five key challenges warrant further investigation.

First, privacy-preserving methods based on cryptographic techniques are predominantly used during the prediction phase and rarely during the training phase. The reasons are as follows:

- Homomorphic encryption (HE) generates larger and more complex ciphertexts. As the number of operations increases, the computational circuit depth deepens, and once it exceeds a certain threshold, decryption may fail to produce correct results.
- Deep learning is inherently a computationally intensive task, requiring significant computational resources and bandwidth overhead. Even without encryption, it necessitates high-throughput computational units.

Given these constraints, the most direct and effective method for preserving privacy involves cryptographic techniques. Therefore, developing efficient cryptographic methods for privacy-preserving machine learning during training remains an urgent problem to address.

Second, many existing cloud platform applications cannot handle encrypted data, necessitating extensive rewrites of these applications. Additionally, technologies used in current secure multi-party computation (SMC) methods, such as HE, Garbled Circuits (GC), and Oblivious Transfer (OT), have inherent limitations. For instance:

- HE is suitable only for a limited range of operations and cannot directly process the non-linear computations inherent in machine learning.
- GC has high computational complexity as each gate in the circuit requires multiple symmetric key operations. This makes it applicable only for two-party or three-party secure computations, limiting its scalability for larger collaborations.
- OT protocols involve expensive public-key operations, rendering them unsuitable for big data applications.

Consequently, designing a generalised privacy-preserving architecture that can accommodate all stages of machine learning while being scalable to big data remains a significant challenge.

Third, existing privacy-preserving mechanisms are primarily designed for structured data, leaving semi-structured and unstructured data vulnerable to privacy breaches. Yet, the majority of big data consists of semi-structured and unstructured formats, with structured data comprising only a small fraction. Recent studies have shown that deep learning methods can automatically collect and process user photos or videos on social networks, achieving astonishing accuracy in object detection and personal identification. Furthermore, model inversion attacks can reconstruct images from facial recognition systems. Traditional privacy-preserving mechanisms are ill-suited for such applications, and even cryptographic methods may still result in privacy leakage. Therefore, safeguarding the privacy of semi-structured and unstructured data without compromising user experience in applications like online social networks represents a promising research direction.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 564-594, 2025
DOI: 10.55214/25768484.v9i1.4184

Fourth, there is an inherent trade-off among the privacy of training data, the efficiency of the model, and its usability in machine learning. For example:

Differential privacy-based defence methods provide high privacy and efficiency but result in reduced usability due to noise perturbations.

- Homomorphic encryption-based methods ensure high privacy but compromise usability due to polynomial approximations during ciphertext computation.
- SMC-based methods strike a balance between privacy and usability but suffer from low efficiency due to extensive participant interactions and high communication overhead. Thus, establishing a multi-dimensional evaluation framework for privacy-preserving mechanisms is essential. Such a framework would model the relationships among privacy, efficiency, and usability under different models and attack scenarios, enabling optimised trade-offs for various application contexts.

Fifth, measuring the privacy leakage risks of machine learning models is a critical aspect of risk assessment frameworks. While some scholars have begun investigating privacy quantification, existing studies remain fragmented and primarily domain-specific, with limited applicability. Moreover, given the myriad factors involved in privacy leakage, a unified model and framework have yet to emerge. Developing standardised metrics for measuring privacy leakage and robust mechanisms for analysing and assessing privacy risks are pressing topics for further research in machine learning.

Driven by the era of big data, the fourth industrial revolution is set to usher in an age of human intelligence, with machine learning becoming an indispensable part of daily life. However, privacy breaches in machine learning pose significant threats. This article summarises and analyses several typical privacy attacks and their defence mechanisms in machine learning. It elucidates the working principles and key features of mainstream privacy-preserving technologies and reviews the latest research achievements in this domain. Privacy breaches and defences in machine learning represent a dynamic battle between attackers and defenders. As technology evolves—particularly with the rise of federated learning and Machine Learning-as-a-Service (MLaaS) models—the methods for attacking model privacy are becoming increasingly diverse, and the challenges of defence are growing ever more complex. Within the inherent trade-offs among data privacy, model efficiency, and usability, developing privacy-preserving methods tailored to specific scenarios that minimise user privacy leakage in machine learning will remain a long-term challenge.

Acknowledgments:

The authors would like to express their gratitude to the anonymous reviewers for their diligent work. Your valuable feedback has contributed significantly to the meaningfulness of this paper, enriching its content and depth. Additionally, we extend our heartfelt thanks to Krirk University in Thailand for providing research funding, which facilitated the completion of this study.

References

[1] P. Ducange, R. Pecori, and P. Mezzina, "A glimpse on big data analytics in the framework of marketing strategies," *Soft Computing*, vol. 22, no. 1, pp. 325–342, 2018. https://doi.org/10.1007/s00500-017-2536-4

[2] Y. Yin, W. Zhang, Y. Xu, H. Zhang, Z. Mai, and L. Yu, "QoS prediction for mobile edge service recommendation with auto-encoder," *IEEE Access*, pp. 62312–62324, 2019. https://doi.org/10.1109/ACCESS.2019.2914737

[3] Google prediction API, "Google prediction API. https://cloud.google.com/prediction," n.d.

[4] M. Amazon, "https://aws.amazon.com/cn/machine-learning/," n.d.

[5] M. Azure, "https://studio.azureml.net/," n.d.

[6] BigML, "https://bigml.com/," n.d.

[7] C. Song, T. Ristenpart, and V. Shmatikov, "Machine learning models that remember too much," in *In: Proc. of the the 2017 ACM SIGSAC Conf. on Computer and Communications Security. New York: ACM, pp. 587–601, 2017. https://doi.org/10.1145/3133956.3134077*, 2017.

[8] G. Ateniese, L. V. Mancini, A. Spognardi, A. Villani, D. Vitali, and G. Felici, "Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers," *International Journal of Security and Networks*, vol. 10, no. 3, pp. 137-150, 2015. https://doi.org/10.1504/ijsn.2015.071829

[9] Y. Rahulamathavan, R.-W. Phan, S. Veluru, K. Cumanan, and M. Rajarajan, "Privacy-preserving multi-class support vector machine for outsourcing the data classification in cloud," *IEEE Trans. on Dependable and Secure Computing*, vol. 11, no. 5, pp. 467–479, 2014. https://doi.org/10.1109/TDSC.2013.51

[10] M. Wilber and T. Boult, "Secure remote matching with privacy: Scrambled support vector vaulted verification (s 2 v 3)," in *In: Proceeding of the the IEEE Workshop on the Applications of Computer Vision. Piscataway: IEEE, pp. 169–176, 2012. https://doi.org/10.1109/WACV.2012.6163018*, 2012.

[11] R. Bost, R. Popa, S. Tu, and S. Goldwasser, "Machine learning classification over encrypted data," in *In: Proc. of the 22nd Annual Network and Distributed System Security Symposium. Rosten: The Internet Society*, 2015, doi: https://doi.org/10.14722/ndss.2015.23241.

[12] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *In: Proceeding of the 22nd ACM SIGSAC Conference. on Computer and Communications Security. New York: ACM, pp. 1322–1333, 2015. https://doi.org/10.1145/2810103.2813677*, 2015.

[13] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, "Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing," in *In: Proceeding of the 23rd USENIX Security Symptoms Berkeley: USENIX Association, pp. 17–32*, 2014.

[14] J. Hayes, L. Melis, G. Danezis, and E. De Cristofaro, "LOGAN: Evaluating privacy leakage of generative models using generative adversarial networks," *arXiv Preprint arXiv:170507663*, pp. 506-519, 2017.

[15] K. Liu, B. Li, and J. Gao, "Generative model: Membership attack, generalization and diversity," *arXiv Preprint arXiv:180509898*, 2018.

[16] Y. Long *et al.*, "Understanding membership inferences on well-generalized learning models," *arXiv Preprint arXiv:180204889*, 2018.

[17] A. Salem, Y. Zhang, M. Humbert, M. Fritz, and M. Backes, "Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models," *arXiv Preprint arXiv:180601246*, 2018.

[18] G. Regulation, "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46. Official Journal of the European Union (OJ)," vol. 59, no. 1-88, p. 294, 2016.

[19] California Consumer Privacy Act (CCPA), "Fines and consumer damages. https://www.clarip.com/data-privacy/california- consumer-privacy-act-fines/," n.d.

[20] China's cyber security law, "http://www.xinhuanet.com/politics/2016-11/07/c_1119867015.htm," 2016.

[21] L. Brandeis and S. Warren, "The right to privacy," *Harvard Law Review*, vol. 4, no. 5, pp. 193-220, 1890.

[22] J. Sarah and C. Melissa, *The international covenant on civil and political rights and United Kingdom law. Int'l covenant on civil and political rights*. Oxford: Clarendon Press, 1995.

[23] J. Saltzer and M. Schroeder, "The protection of information in computer systems," in *Proceeding of the IEEE, vol. 63, no. 9, pp. 1278–1308, 1975. https://doi.org/10.1109/PROC.1975.9939*, 1975.

[24] S. Zhou, F. Li, Y. Tao, and X. Xiao, "Privacy preservation in database applications: A survey," *Chinese Journal of Computers*, vol. 32, no. 5, pp. 847–861, 2009. https://doi.org/10.3724/SP.J.1016.2009.00847

[25] A. Sanyal, M. Kusner, A. Gascon, and V. Kanade, "TAPAS: Tricks to accelerate (encrypted) prediction as a service," *arXiv Preprint arXiv:180603461*, 2018.

[26] P. Xie, M. Bilenko, T. Finley, R. Gilad-Bachrach, K. Lauter, and M. Naehrig, "Crypto-nets: Neural networks over encrypted data," *arXiv Preprint arXiv:14126181*, 2014.

[27] T. Rindfleisch, "Privacy, information technology, and health care," *Communications of the ACM*, vol. 40, no. 8, pp. 92–100, 1997. https://doi.org/10.1145/257874.257896

[28] R. Bolton and D. Hand, "Statistical Fraud detection: A review," *Statistical Science*, vol. 17, no. 3, pp. 235–249, 2002. https://doi.org/10.1214/ss/1042727940

[29] N. Papernot, P. Mcdaniel, I. Goodfellow, S. Jha, Z. Celik, and A. Swami, "Practical black-box attacks against deep learning systems using adversarial examples. ," *arXiv Preprint arXiv:160202697*, vol. 1, no. 2, p. 3, 2016.

[30] P. Laskov, "Practical evasion of a learning-based classifier: A case study," in *In: Proceeding of the 2014 IEEE Symp.on Security and Privacy. IEEE, pp. 197–211, 2014. https://doi.org/10.1109/SP.2014.20*, 2014.

[31] Ú. Erlingsson, V. Pihur, and A. Korolova, "Rappor: Randomized aggregatable privacy-preserving ordinal response," in *In: Proceeding of the 2014 ACM SIGSAC Conference on Computer and Communications Security. 2014. 1054–1067. https://doi.org/10.1145/2660267.2660348*, 2014.

[32] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *In: Proceeding of the 22nd ACM SIGSAC Conference on Computer and Communications Security. New York: ACM, pp. 1310–1321, 2015. https://doi.org/10.1145/2810103.2813687*, 2015.

[33] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Stand-alone and federated learning under passive and active white-box inference attacks," *arXiv Preprint arXiv:181200910*, 2018.

[34] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the GAN: Information leakage from collaborative deep learning," in *In: Proceeding of the ACM SIGSAC Conference on Computer and Communications Security. New York: ACM, 2017. 603–618. https://doi.org/10.1145/3133956.3134012*, 2017.

[35] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Inference attacks against collaborative learning," *arXiv Preprint arXiv: 180504049*, 2018.

[36] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proceeding of the IEEE Symp. on Security and Privacy. Piscataway: IEEE, pp. 3–18, 2017. https://doi.org/10.1109/sp.2017.41*, 2017.

[37] F. Tramèr, F. Zhang, A. Juels, M. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction APIs," in *In: Proceeding of the USENIX Security Symptoms. Berkeley: USENIX Association, 601–618*, 2016.

[38] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *In: Proceeding of the IEEE Symptoms. on Security and Privacy. Piscataway: IEEE*, 2019.

[39] V. Nikolaenko, U. Weinsberg, S. Ioannidis, M. Joye, D. Boneh, and N. Taft, "Privacy-preserving ridge regression on hundreds of millions of records," in *In: Proceeding of the 2013 IEEE Symptoms. on Security and Privacy (SP). IEEE, pp. 334–348, 2013. https://doi.org/10.1109/SP.2013.30*, 2013.

[40] K. Chaudhuri and C. Monteleoni, "Privacy-preserving logistic regression," in *In: Advances in Neural Information Processing Systems. Cambridge: MIT Press, pp. 289–296, 2009. https://doi.org/10.12720/jait.6.3.88-95*, 2009.

[41] Y. Li, Z. Jiang, L. Yao, X. Wang, S. Yiu, and Z. Huang, "Outsourced privacy-preserving C4.5 decision tree algorithm over horizontally and vertically partitioned dataset among multiple parties," *Cluster Computing*, pp. 1–13, 2017. https://doi.org/10.1007/s10586-017-1019-9

[42] S. Samet and A. Miri, "Privacy-preserving back-propagation and extreme learning machine algorithms," *Data & Knowledge Engineering*,, vol. 79, pp. 40–61, 2012. https://doi.org/10.1016/j.datak.2012.06.001

[43] K. Prasad, K. Reddy, and D. Vasumathi, "Privacy-preserving naive Bayesian classifier for continuous data and discrete data," in *In: Proceeding of the 1st Int'l Conference on Artificial Intelligence and Cognitive Computing. Berlin, Heidelberg: Springer-Verlag, pp. 289–299, 2019. https://doi.org/10.1007/978-981-13-1580-0_28*, 2019.

[44] L. Aslett, P. Esperança, and C. Holmes, "Encrypted statistical machine learning: New privacy preserving methods," *arXiv Preprint arXiv:150806845*, 2015.

[45] B. Beaulieu-Jones, Z. Wu, C. Williams, and C. Greene, "Privacy-preserving generative deep neural networks support clinical data sharing," *BioRxiv*, p. 159756, 2017. https://doi.org/10.1101/159756

[46] C. Orlandi, A. Piva, and M. Barni, "Oblivious neural network computing via homomorphic encryption," *EURASIP Journal on Information Security*, vol. 2007, no. 1, pp. 1–11, 2007. https://doi.org/10.1155/2007/37343

[47] E. Hesamifard, H. Takabi, and M. Ghasemi, "CryptoDL: Deep neural networks over encrypted data," *arXiv Preprint arXiv:171105189*, 2017.

[48] N. Phan, X. Wu, H. Hu, and D. Dou, "Adaptive Laplace mechanism: Differential privacy preservation in deep learning," in *In: Proceeding of the IEEE Int'l Conference on Data Mining. Piscataway: IEEE, 2017. 385–394. [doi: 10.1109/ICDM.2017.48]*, 2017.

[49] N. Phan, X. Wu, and D. Dou, "Preserving differential privacy in convolutional deep belief networks," *Machine Learning*, vol. 106, no. 9-10, pp. 1681–1704, 2017. https://doi.org/10.1007/s10994-017-5656-2

[50] T. Adesuyi and B. Kim, "A layer-wise perturbation-based privacy preserving deep neural networks," in *In: Proceeding of the Int'l Conference on Artificial Intelligence in Information and Communication. Piscataway: IEEE, pp. 389–394, 2019. https://doi.org/10.1109/ICAIIC.2019. 8669014*, 2019.

[51] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, "Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy," in *In: Proceeding of the 33rd Int'l Conference on Machine Learning. New York: IMLS, pp. 201–210*, 2016.

[52] K. Huang, X. Liu, S. Fu, D. Guo, and M. Xu, "A lightweight privacy-preserving CNN feature extraction framework for mobile sensing," *IEEE Transaction on Dependable and Secure Computing*, vol. 18, no. 3, pp. 1441-1455, 2019. https://doi.org/10.1109/TDSC.2019.2913362

[53] M. Barni, C. Orlandi, and A. Piva, "A privacy-preserving protocol for neural-network-based computation," in *In: Proceeding of the 8th Workshop on Multimedia and Security. New York: ACM, pp. 146–151, 2006. https://doi.org/10.1145/1161366.1161393*, 2006.

[54] H. Chabanne, A. De Wargny, J. Milgram, C. Morel, and E. Prouff, "Privacy-preserving classification on deep neural network," *IACR Cryptology ePrint Archive*, 2017.

[55] Z. Ma, Y. Liu, X. Liu, J. Ma, and F. Li, "Privacy-preserving outsourced speech recognition for smart IoT devices," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8406–8420, 2019. https://doi.org/10.1109/JIOT.2019.2917933

[56] N. Papernot, M. Abadi, U. Erlingsson, I. Goodfellow, and K. Talwar, "Semi-supervised knowledge transfer for deep learning from private training data," *arXiv Preprint arXiv:161005755*, 2016.

[57] J. Vaidya and C. Clifton, "Privacy-preserving k-means clustering over vertically partitioned data. In: Proceeding of the 9th ACM SIGKDD Int'l Conference on Knowledge Discovery and Data mining." New York: ACM, 2003, pp. 206–215.

[58] Z. Gheid and Y. Challal, "Efficient and privacy-preserving k-means clustering for big data mining," in *In Proceeding of the 2016 IEEE Trustcom/BigDataSE/ISPA. Piscataway: IEEE, pp. 791–798, 2016. https://doi.org/10.1109/TrustCom.2016.0140*, 2016.

[59] Z. Wang, Y. Liu, Z. Ma, X. Liu, and J. Ma, "LiPSG: Lightweight privacy-preserving Q-learning based energy management for the IoT-enable smart grid," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 3935-3947, 2020. https://doi.org/10.1109/JIOT.2020.2968631

[60] P. Trieu, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Trans. on Information Forensics and Security*, vol. 13, no. 5, pp. 1333–1345, 2018. https://doi.org/10.1109/TIFS.2017.2787987

[61] E. Hesamifard, M. Ghasemi, and A. Et, "Privacy-preserving machine learning in cloud. ," *In: Proceeding of the 2017 on Cloud Computing Security Workshop*, pp. 39–43, 2017. https://doi.org/10.1145/3140649.3140655

[62] Q. Zhang, L. Yang, and Z. Chen, "Privacy preserving deep computation model on cloud for big data feature learning," *IEEE Transaction on Computers*, vol. 65, no. 5, pp. 1351–1362, 2016. https://doi.org/10.1109/TC.2015.2470255

[63] N. Chandran, D. Gupta, A. Rastogi, R. Sharma, and S. Tripathi, "EzPC: Programmable, efficient, and scalable secure two-party computation," *IACR Cryptol. ePrint Arch.*, vol. 2017, p. 1109, 2017. https://doi.org/10.1109/eurosp.2019.00043

[64] P. Mohassel and Y. Zhang, "Secure ML: A system for scalable privacy-preserving machine learning," in *Proceeding of the 38th IEEE Symptoms on Security and Privacy. Piscataway: IEEE*, 2017, pp. 19–38, doi: https://doi.org/10.1109/SP.2017.12.

[65] M. S. Riazi, C. Weinert, O. Tkachenko, E. M. Songhori, T. Schneider, and F. Koushanfar, "Chameleon: A hybrid secure computation framework for machine learning applications," in *Proceeding of the Asia Conference on Computer and Communications Security. New York: ACM*, 2018, pp. 707–721, doi: https://doi.org/10.1145/3196494.3196522.

[66] Z. Ma, Y. Liu, X. Liu, J. Ma, and K. Ren, "Lightweight privacy-preserving ensemble classification for face recognition," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5778–5790, 2019. https://doi.org/10.1109/JIOT.2019.2905555

[67] B. Rouhani, M. Riazi, and F. Koushanfar, "Deepsecure: Scalable provably-secure deep learning," in *Proceeding of the 55th ACM/ESDA/I EEE Design Automation Conference Piscataway: IEEE*, 2018, pp. 1–6, doi: https://doi.org/10.1145/3195970.3196023.

[68] C. Juvekar, V. Vaikuntanathan, and A. Chandrakasan, "Gazelle: A low latency framework for secure neural network inference," in *Proceeding of the 27th USENIX Security Symptoms*, 2018, pp. 1651–1669.

[69] J. Liu, M. Juuti, Y. Lu, and N. Asokan, "Oblivious neural network predictions via minionn transformations," in *Proceeding of the 2017 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM*, 2017, pp. 619–631, doi: https://doi.org/10.1145/3133956.3134056.

[70] W. Henecka, A.-R. Sadeghi, T. Schneider, and I. Wehrenberg, "TASTY: Tool for automating secure two-party computations," in *Proceeding ofthe 17th ACM Conference on Computer and Communications Security. New York: ACM*, 2010, pp. 451–462, doi: https://doi.org/10.1145/1866307.1866358.

[71] C. Dwork, "Differential privacy," presented at the In: Encyclopedia of Cryptography and Security, 2011.

[72] Q. Ye, X. Meng, M. Zhu, and Z. Huo, "Survey on local differential privacy," *Ruan Jian Xue Bao/Journal of Software*, vol. 29, no. 7, pp. 1981–2005, 2018. https://doi.org/10.13328/j.cnki.jos.005364

[73] D. Kifer and B. Lin, "Towards an axiomatization of statistical privacy and utility," in *Proceeding of the 29th ACM SIGMOD-SIGACT- SIGART Symptoms on Principles of Database Systems. New York: ACM*, 2010, pp. 147–158.

[74] L. Xie, K. Lin, S. Wang, F. Wang, and J. Zhou, "Differentially private generative adversarial network," *arXivPreprintarXiv:180206739*, 2018. https://doi.org/10.475/123_4

[75] V. Bindschaedler, R. Shokri, and C. Gunter, "Plausible deniability for privacy-preserving data synthesis," *Proceeding of the VLDB Endowment*, vol. 10, no. 5, pp. 481–492, 2017. https://doi.org/10.14778/3055540.3055542

[76] V. Bindschaedler and R. Shokri, "Synthesizing plausible privacy-preserving location traces," in *Proceeding of the 2016 IEEE Symp. on Security and Privacy (SP). Piscataway: IEEE*, 2016, pp. 546–563, doi: https://doi.org/10.1109/SP.2016.39.

[77] M. Liu, H. Jiang, J. Chen, A. Badokhon, X. Wei, and M. Huang, "A collaborative privacy-preserving deep learning system in distributed mobile environment," *In: Proceeding of the Int'l Conference on Computational Science and Computational Intelligence. Piscataway: IEEE*, pp. 192–197, 2017. https://doi.org/10.1109/CSCI.2016.42

[78] M. Abadi *et al. Deep learning with differential privacy*, doi: https://doi.org/10.1145/2976749.2978318.

[79] N. Phan, X. Wu, H. Hu, and D. Dou, "Adaptive Laplace mechanism: Differential privacy preservation in deep learning," in *Proceeding of the IEEE Int'l Conference on Data Mining. Piscataway: IEEE*, 2017, pp. 385–394.

[80] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PloS One*, vol. 10, no. 7, p. e0130140, 2015. https://doi.org/10.1371/journal.pone.0130140

[81] K. Chaudhuri, C. Monteleoni, and A. Sarwate, "Differentially private empirical risk minimization," *Journal of Machine Learning Research*, vol. 12, pp. 1069–1109, 2011. https://doi.org/10.1109/MIS.2011.2

[82] C. Dwork, F. Mcsherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proceeding of the Theory of Cryptography Conference Berlin, Heidelberg: Springer-Verlag*, 2006, no. 265–284, doi: https://doi.org/10.1007/11681878_14.

[83] N. Phan, Y. Wang, X. Wu, and D. Dou, "Differential privacy preservation for deep auto-encoders: An application of human behavior prediction," in *In: Proceeding of the 30th AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press*, 2016, pp. 1309–1316.

[84] B. Jayaraman, L. Wang, D. Evans, and Q. Gu, "Distributed learning without distress: Privacy-preserving empirical risk minimization," *In: Advances in Neural Information Processing Systems*, pp. 6343–6354, 2018.

[85] N. Papernot, S. Song, I. Mironov, A. Raghunathan, K. Talwar, and Ú. Erlingsson, "Scalable private learning with PATE," *arXiv Preprint arXiv:180208908*, 2018.

[86] I. Mironov, "Rényi differential privacy," in *Proceeding of the 30th IEEE Computer Security Foundations Symp. (CSF). Piscataway: IEEE*, 2017, pp. 263–275, doi: https://doi.org/10.1109/CSF.2017.11.

[87] X. Wu, F. Li, A. Kumar, K. Chaudhuri, S. Jha, and J. Naughton, "Bolt-on differential privacy for scalable stochastic gradient descent-based analytics," in *Proceeding of the 2017 ACM Int'l Conference on Management of Data*, 2017, pp. 1307–1322, doi: https://doi.org/10.1145/3035918.3064047.

[88] B. Jayaraman and D. Evans, "Evaluating differentially private machine learning in practice," presented at the 28th USENIX Security Symposium (USENIX Security 19), 2019.

[89] T. Graepel, K. Lauter, and M. Naehrig, "ML confidential: Machine learning on encrypted data," in *Proceeding of the Int'l Conference on Information Security and Cryptology. Berlin, Heidelberg: Springer-Verlag, pp. 1–21. https://doi.org/10.1007/978-3- 642-37682-5_1*, 2012.

[90] Z. Li, X. Gui, Y. Gu, X. Li, H. Dai, and X. Zhang, "Survey on homomorphic encryption algorithm and its application in the privacy-preserving for cloud computing," *Ruan Jian Xue Bao/Journal of Software*, vol. 29, no. 7, pp. 1827–1851, 2018. https://doi.org/10.13328/j.cnki.jos.005354

[91] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes: Theory and implementation," *ACM Computing Surveys*, vol. 51, no. 4, pp. 1-35, 2018. https://doi.org/10.1145/3214303

[92] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceeding of the 41st Annual ACM Symp. on Theory of Computing,, New York: ACM* 2009, p. 169−178, doi: https://doi.org/10.1109/TIFS.2013.2287732.

[93] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) fully homomorphic encryption without bootstrapping," *ACM Transactions on Computation Theory*, vol. 6, no. 3, pp. 1-36, 2014. https://doi.org/10.1145/2090236.2090262

[94] A. El-Yahyaoui and M. D. E.-C. El Kettani, "An Efficient Fully Homomorphic Encryption Scheme," *IJ Network Security*, vol. 21, no. 1, pp. 91-99, 2019. https://doi.org/10.6633/IJNS.20190121(1).11

[95] Y. Ichibane, Y. Gahi, M. Guennoun, and Z. Guennoun, "Fully Homomorphic Encryption Without Noise," *International Journal of Smart Security Technologies*, vol. 6, no. 2, pp. 33-51, 2019. https://doi.org/10.4018/IJSST.2019070102

[96] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène, "TFHE: Fast fully homomorphic encryption over the torus," *Journal of Cryptology*, vol. 33, no. 1, pp. 34-91, 2020.

[97] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or—1," *arXiv Preprint arXiv:160202830*, 2016.

[98] M. Kim and P. Smaragdis, "Bitwise neural networks," *arXiv Preprint arXiv:160106071*, 2016.

[99] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachene, "Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds," in *In: Proceeding of the the Int'l Conference on the Theory and Application of Cryptology and Information Security. Berlin, Heidelberg: Springer- Verlag, pp. 3–33. https://doi.org/10.1007/978-3-662-53887-6_1*, 2016.

[100] F. Bourse, M. Minelli, M. Minihold, and P. Paillier, "Fast homomorphic evaluation of deep discretized neural networks," in *In: Proceeding of the Annual Int'l Cryptology Conference Berlin, Heidelberg: Springer-Verlag, pp. 483–512, 2018. https://doi.org/10.1007/978-3-319-96878-0_17*, 2018.

[101] M. Baryalai, J. Jang-Jaccard, and D. Liu, "Towards privacy-preserving classification in neural networks," in *Proceeding of the 14th Annual Conference on Privacy, Security and Trust (PST), IEEE*, 2016, pp. 392–399,, doi: https://doi.org/10.1109/PST.2016.7906962.

[102] M. Li, S. Chow, S. Hu, ., Y. Yan, M. Du, and Z. Wang, "Optimizing privacy-preserving outsourced convolutional neural network predictions," *arXiv Preprint arXiv:200210944*, 2020.

[103] L. Liu *et al.*, "Toward highly secure yet efficient KNN classification scheme on outsourced cloud data," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9841-9852, 2019. https://doi.org/10.1109/JIOT.2019.2932444

[104] J. Bos, K. Lauter, J. Loftus, and M. Naehrig, "Improved security for a ring-based fully homomorphic encryption scheme," *In: Proc. of the IMA Int'l Conference on Cryptography and Coding. Berlin, Heidelberg: Springer-Verlag*, pp. 45–64, 2013. https://doi.org/10.1007/978-3-642-45239-0_4

[105] A. Bansal, T. Chen, and S. Zhong, "Privacy preserving back-propagation neural network learning over arbitrarily partitioned data," *Neural Computing and Applications*, vol. 20, no. 1, pp. 143–150, 2011. https://doi.org/10.1007/s00521-010-0346-z

[106] S. Mehnaz, G. Bellala, and E. Bertino, "A secure sum protocol and its application to privacy-preserving multi-party analytics," in In: Proceeding of the 22nd ACM on Symptoms on Access Control Models and Technologies. New York: ACM, pp. 219–230, 2017. https://doi.org/10.1145/3078861.3078869, 2017.

[107] S. Abbasi, S. Cimato, and E. Damiani, "Toward secure clustered multi-party computation: A privacy-preserving clustering protocol," in *Proceeding of the Information and Communication Technology-Eur Asia Conference Berlin, Heidelberg: Springer-Verlag*, 2013, pp. 447–452, https://doi.org/10.1007/978-3-642-36818-9_49.

[108] G. Asharov, Y. Lindell, T. Schneider, and M. Zohner, "More efficient oblivious transfer extensions," *Journal of Cryptology*, vol. 30, pp. 805-858, 2017. https://doi.org/10.1007/s00145-016-9236-6

[109] V. Dani, V. King, M. Movahedi, J. Saia, and M. Zamani, "Secure multi-party computation in large networks," *Distributed Computing*, vol. 30, pp. 193-229, 2017. https://doi.org/10.1007/s00446-016-0284-9

[110] D. Bogdanov, M. Niitsoo, T. Toft, and J. Willemson, "High-performance secure multi-party computation for data mining applications," *International Journal of Information Security*, vol. 11, pp. 403-418, 2012. https://doi.org/10.1007/s10207-012-0177-2

[111] P. Mohassel, Y. Zhang, and M. Secure, "A system for scalable privacy-preserving machine learning," *Proceeding of the 38th IEEE Symptoms on Security and Privacy. Piscataway: IEEE, pp. 19–38*, 2017. https://doi.org/10.1109/SP.2017.12

[112] A. Gascón *et al.*, "Privacy-preserving distributed linear regression on high-dimensional data," *Proceeding on Privacy Enhancing Technologies*, vol. 2017, no. 4, pp. 345–364, 2017. https://doi.org/10.1515/popets-2017-0053

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 1: 564-594, 2025
DOI: 10.55214/25768484.v9i1.4184