Efficiency Optimization Techniques in Privacy-Preserving Federated Learning With Homomorphic Encryption: A Brief Survey

Qipeng Xie, Siyang Jiang, Linshan Jiang, *Member, IEEE*, Yongzhi Huang, Zhihe Zhao, Salabat Khan, *Member, IEEE*, Wangchen Dai, Zhe Liu, *Senior Member, IEEE*, and Kaishun Wu, *Fellow, IEEE*

***Abstract*—Federated learning (FL) offers distributed machine learning on edge devices. However, the FL model raises privacy concerns. Various techniques, such as homomorphic encryption (HE), differential privacy, and multiparty cooperation, are used to address the privacy issues of the FL model. Among them, HE ensures greater security and privacy since end-to-end encryption maintains data privacy throughout the computation process. Compared with other privacy-preserving techniques, HE does not require the establishment of a trusted environment or protocol among multiple parties and does not involve any artificial noise that can impair system performance. Unfortunately, it suffers from efficiency overhead when applied to privacy-preserving FL (PPFL). Some existing surveys on PPFL discuss the generic construction and organization of PPFL from the perspective of practical HE deployment in PPFL. However, none of them covers the efficiency optimization of HE when applied to PPFL. This article conducts a comprehensive review of the efficiency optimization of HE when applied to PPFL. First, we review general optimization strategies and discuss their limitations when applied directly to HE-based PPFL. Second, an overview of algorithmic, hardware, and hybrid optimizations is provided, along with a discussion of their adaptation. Additionally, we provide a detailed taxonomy of optimizations. Finally, we suggest future HE-based PPFL research directions.**

***Index Terms*—Efficiency optimization, federated learning (FL), homomorphic encryption (HE), Internet of Things (IoT), privacy.**

Manuscript received 25 October 2023; revised 27 February 2024; accepted 20 March 2024. Date of current version 9 July 2024. This work was supported in part by China, NSFC under Grant U2001207; in part by the Guangdong Provincial Key Lab of Integrated Communication, Sensing and Computation for Ubiquitous Internet of Things under Grant 2023B1212010007; in part by the Project of DEGP under Grant 2023KCXTD042; in part by the Zhejiang Lab Open Research Project under Grant K2022PDOAB01; in part by the Foundation for Distinguished Young Talents in Higher Education of Guangdong Province, China, under Grant 2022KQNCX084; and in part by the Fund which aims to improve scientific research capability of key construction disciplines in Guangdong province "Light-Weight Federal Learning Paradigm and Its Application" under Grant 2022ZDJS058. *(Corresponding authors: Salabat Khan; Kaishun Wu.)*

Qipeng Xie is with the Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China, also with The Hong Kong University of Science and Technology, 511453 Hong Kong, and also with the Zhejiang Laboratory, Hangzhou 311121, China (e-mail: qxieaf@connect.ust.hk).

Siyang Jiang is with the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong, SAR, and also with the Zhejiang Laboratory, Hangzhou 311121, China (e-mail: syjiang@ie.cuhk.edu.hk).

Linshan Jiang is with the Institute of Data Science, National University Singapore, Singapore 119077 (e-mail: linshan@nus.edu.sg).

Yongzhi Huang and Kaishun Wu are with the Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China (e-mail: huangyongzhi@email.szu.edu.cn; wuks@hkust-gz.edu.cn).

Zhihe Zhao is with the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong, SAR.

Salabat Khan is with the School of Computer and Information Engineering, Qilu Institute of Technology, Jinan 250202, China (e-mail: salabatwazir@gmail.com).

Wangchen Dai and Zhe Liu are with the Zhejiang Laboratory, Hangzhou, China (e-mail: w.dai@my.cityu.edu.hk; zhe.liu@nuaa.edu.cn).

Digital Object Identifier 10.1109/JIOT.2024.3382875

I. Introduction

THE Internet of Things (IoT) devices revolutionize various sectors such as healthcare, agriculture, finance, and industries by enhancing efficiency and productivity [1], [2]. IoT devices allow real-time monitoring of patient health [3], [4], irrigation systems in agriculture, smart financial systems, and streamlined industrial operations. The enormous data generated by IoT devices at the network edge are driving the deployment of machine learning algorithms at the network edge, called edge learning [5], to distill the data into intelligence. This intelligence can then be used to support AI-powered applications, ranging from virtual reality to e-commerce [6].

One of the most popular frameworks for distributed machine learning on edge devices is federated learning (FL) [7], [8]. FL provides a novel paradigm by allowing decentralized model training through local model aggregation to protect the privacy of raw data [7], [9]. In representative FL frameworks like FedAvg [10], and FedBE [11], rather than uploading raw data to a central server for training, clients train local models with their own data and uploading local models to the server. Then, the server aggregates the uploaded model to learn a high-quality global model.

However, FL schemes expose individual local models to the aggregation server, which may maintain the privacy-sensitive information of local data. Such uploaded models may be eavesdropped and revealed to potential malicious clients in FL systems. Therefore, various advanced privacy attacks [12], [13], [14], [15], [16], [17], [18] have been proposed, allowing the curious server and malicious clients to extract privacy-sensitive information in the training data, including membership inference attacks [12], [13], [14], inversion attacks [15], [16], [17] and private attribute extraction attacks [18].

To defend against the aforementioned attacks, privacy-preserving FL (PPFL) is proposed. In particular, existing defenses in PPFL to counteract the privacy attacks include multiparty computation (MPC) [19], [20] and differential privacy (DP) [21], [22]. MPC enables the involved parties to jointly compute a mutual function to aggregate the local models, i.e., garbled circuit (GC) [23], Masking [19] and secret sharing (SS) [24], [25]. In this case, the local models are securely aggregated while preventing information leakages from the uploaded models. In DP-based PPFL, it conventionally adds noise to sensitive information (e.g., uploaded gradient [26]) to protect the privacy of training data. However, these approaches have some drawbacks. MPC requires extra interactive synchronization steps and is sensitive to the system robustness in FL, caused by issues such as client dropout, necessitating meticulous engineering during implementation. Meanwhile, DP typically leads to a nonnegligible model performance downgrade due to the incremental noises [27], [28].

Another line of study of existing defense in PPFL is homomorphic encryption (HE) [29], [30], [31], [32], which offers a robust post-quantum solution that counteracts privacy attacks while maintaining the full utility of aggregated models. Note that the Paillier cryptosystem is not quantum-resistant because it relies on the hardness of integer factorization, which can be broken in polynomial-time with Shor's algorithm [33]. HE schemes such as CKKS, BGV, and BFV are indeed designed with quantum resistance in mind, as they are based on the hardness of (Ring) learning with errors (LWE) problems. In HE-based FL, clients encrypt local models, and the server performs model aggregation over ciphertexts during aggregation. For example, Mandal and Gong [34] created robust and secure training protocols for federated regression models utilizing HE, which are optimized for IoT devices.

Compared to MPC and DP, HE can be easily adapted to provide strong privacy guarantees without algorithm modifications or accuracy loss. Thus, the approach built upon HE has been a promising solution in PPFL. These HE-based PPFL approaches can enable secure FL deployments and have been adopted by several PPFL systems [35], [36], [37] and domain-specific applications [38], [39].

Currently, several existing surveys in [40], [41], [42], [43], [44], and [45] extensively analyze privacy and security threats to FL systems with discussions on potential attacks and defenses. However, few existing surveys on PPFL perceive the construction and optimization of PPFL from the perspective of efficiency, especially for the deployment of HE on edge devices. For example, the surveys in [40] and [41] extensively analyze the privacy and security threats to FL systems. Furthermore, the surveys in [42] and [43] systematically investigate the current work of vertical FL (VFL) from a layered perspective and provide an overview of the current progress in VFL, respectively. The surveys [44], [45] have surveyed some privacy-preserving aggregation protocols. However, the surveys [44], [45] only briefly discuss efficiency techniques, lacking the depth required for a comprehensive understanding. Several other works [28], [46], [47], [48] either focus on optimizing the performance and efficiency of standard FL [46] or focus on optimizing HE within centralized privacy-preserving machine learning [47], [48] or its simplistic integration into FL without optimization [28]. The comparison of our survey with existing surveys is shown in Table I.

In summary, none of them provides an extensive discussion of the limitations and adaptation of optimization techniques in HE-based PPFL on edges. Consequently, the field of optimization for HE-based FL remains relatively unexplored, lacking in-depth research and consensus among researchers. This motivates us to deliver the survey with a comprehensive review of the literature on efficiency optimization in PPFL from the perspective of HE. We expect this work to be an initial attempt to bridge the gap of efficiency-oriented surveys in the PPFL domain, as well as to solicit more contributions to related research.

The contributions of this survey can be summarized as follows.

1) We provide a detailed overview of HE-based PPFL by starting with a comprehensive discussion on the background of FL, HE, related privacy-preserving techniques (e.g., DP and MPC) in PPFL, and the threat model.
2) We provide a taxonomy of HE-based PPFL schemes (see Fig. 1) and discuss the strengths and weaknesses of each scheme. In addition, we discuss the overall limitations and adaptation of each class.
3) Finally, we present the challenges faced by HE-based PPFL and discuss the potential future research directions of HE-based PPFL.

The rest of this article is organized as follows. Section II introduces the background and preliminary details of FL and HE with some other alternative privacy-preserving techniques. Section III presents the algorithmic optimization techniques that are tailored for HE-based FL. Section IV presents the hardware optimization techniques. Section V presents the hybrid optimization, and Section VI concludes this article.

II. Background and Preliminary

In this section, we provide a brief overview of the state-of-the-art of FL and its categories. We introduce some preliminaries of HE and conventional HE-based PPFL. Besides, we introduce some other alternative privacy-preserving techniques that are feasible in PPFL.

A. Federated Learning Overview

There are two roles in a conventional FL setting. The one is a set of n clients C, in which each client C_i, $i \in (1, \ldots, n)$, has a local training data set D_i and trains their local model θ_i. The other is an aggregation server S that receives and aggregates model θ_i updates from clients and computes a global model θ without accessing the client's raw data.

The FL model and the traditional centralized learning model, obtained through the federated and traditional training process, are defined as θ and θ_{cen}, respectively. Due to the existence of parameter exchange and aggregation operations, there may be a loss of accuracy throughout the training process, that is, the performance of θ is not as good as that of θ_{cen}. To quantify this difference, the performance of θ on the test set is denoted as *per*, and the performance of θ_{cen} on the

TABLE I
COMPARISON OF OUR SURVEY WITH EXISTING SURVEYS

Ref	Published on	Contribution	Optimization for HE-based PPFL?
[37]	IEEE TNNLS 2022	Privacy and security threats of FL	No
[38]	ACM Survey 2021	Privacy and security threats of FL	No
[40]	ACM Survey 2023	Overview of the VFL	Few optimization for HE-based PPFL
[41]	PETS 2023	Privacy-preserving aggregation	Few optimization for HE-based PPFL
[42]	IEEE TBD 2022	Privacy-preserving aggregation	Few optimization for HE-based PPFL
[43]	IEEE TBD 2022	Overview of the FL optimization	Only for FL optimization
Ours		Overview of HE-based PPFL for efficiency optimization	Comprehensive optimization for HE-based PPFL

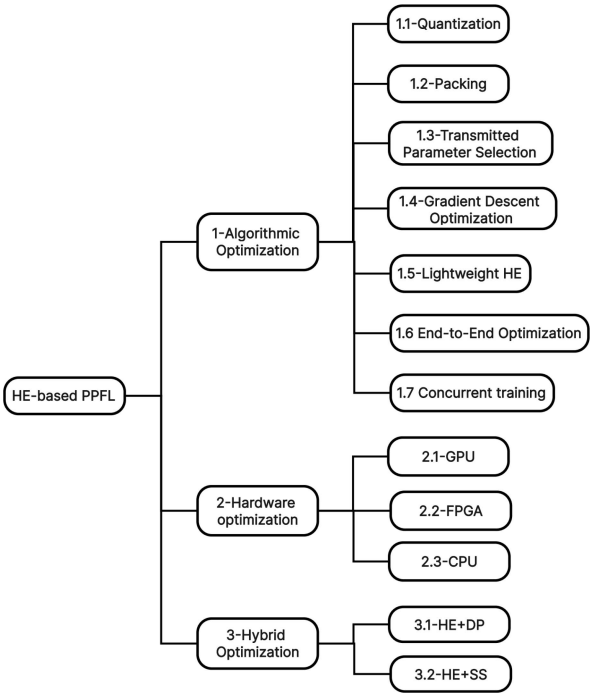

Fig. 1. Taxonomy of HE-based PPFL optimizations.

test set is denoted as per_{cen}. The δ-accuracy loss of the model is then defined as: $|per - per_{\text{cen}}| < \delta$.

B. Federated Learning Categories

According to the recent advances in FL, we hereby classify FL from the data partitioning perspective as follows.

Data Partitioning: This class can be further divided into horizontal FL (HFL) and VFL based on the distribution of training data over the samples and features [49], as described in Table II.

1) *HFL:* HFL refers to the FL setting where participants share the same feature space while holding different samples. For example, Google uses HFL to allow mobile phone users to use their data set to collaboratively train a next-word prediction model [10]. In this case, users have different sample spaces but the same feature space, and each user owns the labels of their samples.
2) *VFL:* VFL refers to the FL setting where data sets share the same samples while having different features. For example, multiple medical institutions, e.g., hospitals, own different records for the same patients [50], [51]. They could securely train better models for healthcare tasks with VFL. In this case, users have the same sample space but different feature spaces. Only one participant holds the label of the FL task. Before training a model, all participants have to align the samples among different data silos based on the common users.

FL can also be categorized into "cross-device" and "cross-silo" settings [52]. The cross-device FL setting typically involves a large number of IoT devices as the participating users, whereas the cross-silo FL setting involves a limited number of organizations. HFL can be applied to either cross-device or cross-silo FL settings, while VFL is more commonly associated with cross-silo FL. The division of cross-silo and cross-device is not a key challenge for HE-based FL; thus, we mainly discuss their application in the corresponding works.

C. Homomorphic Encryption Overview

HE allows arithmetic computations on encrypted data without decryption. It generally consists of four components, including KeyGen, Encrypt, Decrypt, and Evaluation (i.e., HomoAdd or HomoMult) as follows:

1) *KeyGen:* Generates key pairs (pk, sk).
2) *Encrypt:* Encrypts the message m, producing ciphertext c.
3) *Evaluation:* Applies function f on encrypted data, yielding encrypted result c_{result}.
4) *Decrypt:* Decrypts c_{result} to obtain the plaintext result m_{result}. If designed correctly, $m_{\text{result}} = f(m)$.

Generally, HE schemes can be classified as *partially HE* (PHE) and *fully HE* (FHE) based on the number and types of arithmetic operations permitted on encrypted data.

PHE schemes allow an unlimited number of operations but only support one type of operation, either addition or multiplication. FHE schemes permit an unlimited number of

TABLE II
COMPARISON OF MAIN CHARACTERISTICS BETWEEN CONVENTIONAL HFL AND VFL

Setting	Data partitioning	Scale	Scenario	Exchanged Level	Dropout	Privacy
HFL	Sample space	Cross-devices	IoT devices	Client Gradients	✓	Low
		Cross-silo	Organizations		✗	High
VFL	Feature space	Cross-silo	Organizations	Intermediate Results	✗	High

arithmetic operations and support multiple types of operations, such as addition and multiplication, without any restrictions. This enables the execution of complex functions and algorithms directly on encrypted data.

1) PHE: The most famous approach, Paillier [53], is widely adopted in FL to enable aggregation over encrypted data, thus protecting user privacy.

2) FHE: To protect the privacy of the whole FL workflow, e.g., the global model, FL clients need to train their local model based on an encrypted global model, which requires complicated function evaluation over ciphertext. Therefore, FHE schemes are preferable option because they support addition and multiplication, enabling them to perform arbitrary complex computations on encrypted data.

To the best of our knowledge, most FHE schemes in use are based on the ring LWE (RLWE) problem [54]. These schemes can be classified as word-wise FHE (second-generation and fourth-generation FHE) and bit-wise FHE (third-generation FHE) according to the type of data and basic operations. The first class contains BGV [55], BFV [56] and CKKS [30], where BGV and BFV perform exact operations on integers, and CKKS supports approximate computations over real and complex numbers. The second class, such as FHEW [57] and TFHE [58], encrypts a few bits per ciphertext and performs logical operations. CKKS has the advantages of supporting real number arithmetic processing and supporting single instruction multiple data (SIMD), which is prominent in the fields of privacy-preserving machine learning.

D. Conventional HE-Based PPFL

The workflow of conventional HE-based FL without efficiency optimization is shown in Fig. 2.

1) HE-Based HFL: In HE-based HFL, the goal is to ensure that no client reveals its model updates during aggregation. Several approaches [28], [49], [59] have been proposed to achieve this, including the use of PHE schemes, such as Paillier [53]. With HE, gradient aggregation can be performed on ciphertexts without decrypting them in advance.

In a HE-based HFL setup, a HE key pair is securely shared among clients. Note that if clients share the same pair of HE keys, it could suffer from several privacy risks. This is precisely the issue that the proposed xMK-CKKS [60] scheme aims to address and is outside our scope. During training, clients encrypt their gradient updates with the public key and send the ciphertexts to a central server. The server aggregates the encrypted gradients and returns the result to the clients. Clients decrypt the aggregated gradients using the private key, update their local models, and proceed to the next iteration. Uploading only encrypted updates ensures privacy protection against the server and external parties during data transfer and aggregation.

2) HE-Based VFL: Distinct from horizontal partitioned data, in VFL, participating clients accomplish model training interactively by performing computations and exchanging encrypted intermediate results without revealing original data [49], [61], [62].

The HE-based VFL consists of two parts. The first part is encrypted entity alignment, which identifies common data samples across multiple clients without disclosing nonmatching data. The second part is encrypted model training, which focuses on training a model on the aligned data, considering that the feature spaces of all parties may not be the same, which is the focus of our work. In a given iteration, clients first secure their intermediate results using their respective public keys. After this encryption step, they proceed to share these encrypted results with each other. This exchange enables each client to accumulate a fully encrypted loss that encompasses both the feature data from other participants and their respective label information. With this encrypted loss at hand, clients then calculate the encrypted gradients and transmit these to the server for the decryption process. Once decrypted by the server, the gradients are then distributed back to each client individually. For example, in vertically federated linear models, the gradients are divided into local terms that can be computed at each client and cross terms that can be computed by sending the encrypted intermediate results from one participating client to another [49], [61]. For vertically federated XGBoost models, it is critical to compute the gain and weight for each possible split point of data provider without revealing labels. The work of SecureBoost [62] found that both gain and weight are functions of aggregated gradients g and h, and thus proposed to interactively compute aggregated gradients with additively HE (AHE) to build vertical federated XGBoost models.

However, these conventional HE-based PPFL systems show inefficient system optimization, especially in communication and computation. For example, using Paillier dramatically increases training time [63] ($96\times$, $135\times$) and data transfer compared to updating raw plaintext for FMNIST, CIFAR, respectively. This motivates us to investigate efficiency optimization in HE-based PPFL.

E. Other Privacy-Preserving Techniques in PPFL

DP [64] is a widely adopted privacy-preserving technique in both academia and industry. The basic idea is to add noise to personal sensitive attributes to protect privacy. In

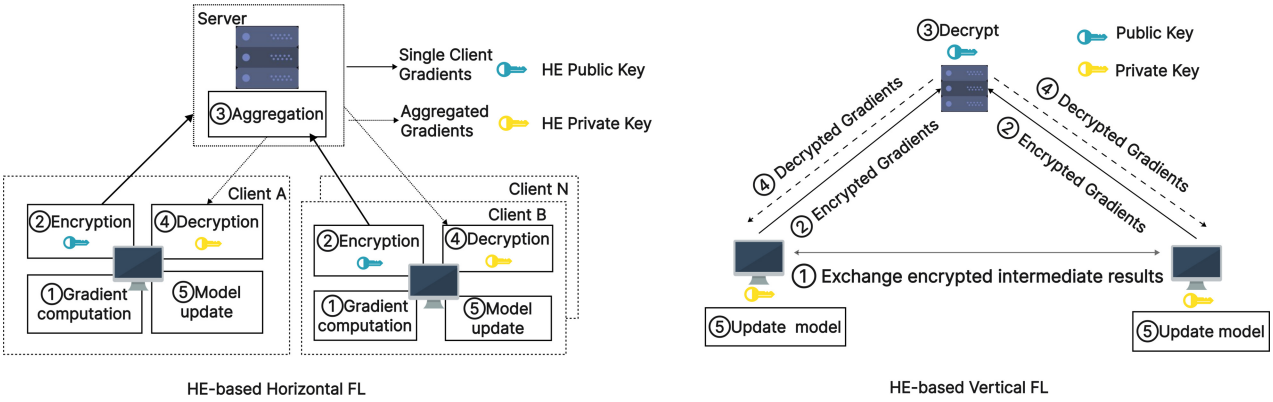

Fig. 2. Conventional HE-based FL without efficiency optimization.

the context of FL, clients add noise to their uploaded model updates to mitigate inference attacks. Note that using DP improves clients' privacy protection at the cost of global model performance (e.g., slower convergence and/or loss of accuracy). Such performance degradation may be a major concern in certain cross-silo settings where there is a strict requirement for model performance.

MPC [65] provides a generic approach that enables clients to jointly compute arbitrary functionality without revealing their raw data. The aggregated model is calculated by exchanging these secret shares among clients following well-designed protocols. SS is indeed a fundamental primitive that lies at the core of many MPC approaches. Informally, a (t, n)-SS scheme divides the secret s into n shares in such a way that any $t-1$ of these shares reveal no information about the secret s. However, any t shares allow for a complete reconstruction of the secret s.

F. Threat Model

In constructing a threat model for FL, it is essential to assess the capabilities of potential adversaries to ensure that the chosen privacy-preserving methods are aptly robust. This article concentrates on two predominant threats:

Semi-Honest Adversaries: These participants comply with established protocols, yet possess an ulterior motive to infringe upon the confidentiality of the data communicated. Their strategy is to passively glean sensitive information without outwardly tampering with the protocol's operation.

Malicious Adversaries: Unlike their semi-honest counterparts, these actors willfully breach protocol guidelines with the explicit intent to deceive or undermine the system's security. Their methods are active, involving direct interference in the protocol's execution to manipulate outcomes in their favor.

Besides, such adversaries may represent individual clients or the central server coordinating the learning process, while they may collude with others. For the noncolluding adversaries, these adversaries act independently, mounting isolated attacks without the support or cooperation of others. In contrast, colluding adversaries coordinate and conspire, potentially leading to more sophisticated and damaging assaults on privacy or the integrity of the global model. For example, semi-honest colluding clients may collude with the server to try to steal sensitive information from other honest clients. In this survey, we delve into detailed analyses of these adversarial models and discuss the implications for the robustness of privacy-preserving mechanisms within HE-based PPFL frameworks.

III. Algorithmic Optimization for Practical HE-Based PPFL

In recent years, some research efforts have focused on tackling the challenges of fully unleashing the performance potential of HE-based PPFL. We mainly review them from three aspects: algorithmic optimization (see Section III), hardware optimization (see Section IV), and hybrid optimization (see Section V) with other privacy-preserving techniques. Note that some works apply several types of optimization concurrently. Table III summarizes the optimizations for practical HE-based PPFL.

As mentioned above, this section provides a summary of the algorithmic optimization techniques utilized in HE-based PPFL. Specifically, we have categorized these optimization techniques into seven categories: quantization, packing, transmitted parameter selection (TPS), gradient descent optimization (GDO), lightweight HE, end-to-end optimization, and concurrent training owing to the different phases in FL.

A. Model Transmission Phase

1) Quantization: Quantization methods in FL aim to minimize network communication costs by transforming high-precision gradients into low-precision numbers while retaining critical information. For example, Batchcrypt [63] is an innovative framework that synergistically combines quantization, batch encoding, and analytical quantization modeling to improve computational speed and communication efficiency without compromising model quality. In particular, the authors initially proposed an analytical model, called dACIQ, which clips gradients to reduce error. Next, they quantize the clipped gradient values into uniformly distributed signed integers within a symmetric range. They also devised a novel batch encoding scheme employing two's complement representation with two sign bits for well-quantized values while using padding and advanced scaling to avoid overflow in aggregation.

Inspired by Batchcrypt [63], SecureBoost+ [66] first encodes encrypted first-order and second-order gradients into

TABLE III
OPTIMIZATIONS FOR PRACTICAL HE-BASED PPFL

Work	Algorithmic	Hardware acceleration	Hybrid	Threat Model	Strength	Weakness
Aono et at. [25]	Packing	✗	✗	Semi-honest & Non-colluding	It enhances privacy using HE without sacrificing model accuracy	High communication overhead impacting scalability
Batchcrypt [60]	Quantization, Packing	✗	✗		Reduces training time and communication cost	Not suitable for VFL
Secure Boost+ [63]	Packing, Concurrent training	✗	✗		Optimizes ciphering and training	Not suitable for HFL
eHE-SecureBoost [64]	Quantization, Packing	✗	✗		Quantizes gradients to unsigned binary numbers suitable for VFL	Not suitable for HFL
Vf2boost [73]	Packing, Concurrent training	✗	✗		Customized HE operations Reduce the idle periods	Not suitable for HFL
Eastfly [66]	Quantization, Packing	✗	HE+SS	Semi-honest & Colluding	Quantizes gradients to tenary vector	Accuracy decrease
Zhang et al [72]	Packing	✗	HE+Bilinear	Malicious & Non-colluding	CRT packing and verifiability	Verifiability brings latency
Esafl [74]	Packing	✗	HE+SS	Semi-honest & Colluding	It does not rely on secure channel for ciphertext transmission	One-step secure decryption reduces security
Li et al. [75]	TPS GDO	✗	✗	Semi-honest & Non-colluding	Asynchronous VFL and double-end sparsification	Not suitable for HFL
AVFL [78]	TPS	✗	✗		PCA to perform dynamic feature selection	Multiplication unfriendly encryption strategy
FLARE [32]	TPS	✗	✗		NVIDIA, performs HE operations on certain layers	Loss of accuracy
FedML-HE [82]	TPS	✗	✗		Parameter-wise selection for more fine-grained overhead control	No specific optimization for HE
Liu et al. [83]	GDO	✗	✗		Federated stochastic block coordinate descent algorithm (FeDBCD)	Complex and asynchronized collaborative systems
Yang et al. [87]	GDO	✗	✗		Quasi-Newton method for fast convergence	No specific optimization for HE
ACML * [85]	GDO	✗	✗		HE-based backpropagation algorithm	No specific optimization for HE
VPFL [88]	GDO	✗	HE+Bilinear	Malicious & Non-colluding	DSSGD-based online/offline signature support and prevent gradients leakage	No specific optimization for HE
Hao et al. [90]	Lightweight HE	✗	HE+SS+DP	Semi-honest & Colluding	Symmetric HE	Symmetric HE Reduces security
HAFLO [95]	✗	GPU	✗	Semi-honest & Non-colluding	GPU-based solution for accelerating FLR	Only implemented LR
Carm * [96]	✗	GPU	✗		First using GPU for accelerating FHE in IOT	Have not implemented in FL
Yang et al [97]	✗	FPGA	✗		Strategically offloaded the modular multiplication	Not efficient enough
Flash [98]	✗	FPGA	✗		Thorough analysis towards all cryptographic operations used in cross-silo FL	Only for cross-silo FL
Fate [99]	✗	CPU	✗		HEXL, webank	Not efficient enough
SPINDLE [93]	End-to-End LR	✗	✗	Semi-honest & Colluding	First PPFL system covers the complete FL workflow	Only implemented LR
POSEIDON [94]	End-to-End NN	✗	HE+SS		First PPFL system with multiple parties for NN	Lacks gradient clipping
CryptoBoost [92]	End-to-End Xgboost	✗	HE+SS		First PPFL system with N parties for Xgboost	Lack of robust security analysis
Kim et al. [100]	✗	✗	HE+DP	Semi-honest & Non-colluding	First work that combines DP and HE in PPFL	Lacks generality
Chai et al. [101]	✗	✗	HE+DP	Semi-honest & Non-colluding	Efficient federated matrix factorization method	Lacks generality
Pivot [102]	✗	✗	HE+SS	Semi-honest & Colluding	First work that combines SS and HE in PPFL	Loss of accuracy
Privfl [31]	Lightweight HE	✗	HE+SS		Can be used in edge edge devices	Focuses on linear models in HFL
CAESAR [103]	GDO	✗	HE+SS		Build a secure large-scale sparse LR	Only for sparse LR

*Note that the works are not designed FL specifically however they can be naively adopted and transferred in the FL scenario. Thus, we analyze the transferred version in FL.

a single message by adopting a simple fix-point encoding strategy to transform a floating number into a large integer. Meanwhile, eHE-SecureBoost [67] is a novel batch method that shifts all to nonnegative numbers, truncates them, quantizes them to unsigned binary numbers, and puts them together as a batched number for further encryption, transmission, and computations. Han and Yan [68] first go through the Shifting Gradients to Nonnegative Values step to avoid possible overflow caused by the addition of negative numbers represented in two's complement. Then, they apply the binary conversion of a nonnegative floating point Number to ensure the HE additivity of batched ciphertext.

Other studies based on quantization techniques use ternary or binary weight networks to improve computation and communication efficiency. These methods quantize the gradients to ternary {–1, 0, 1} or binary {–1, 1} values, which reduces the size of the data for transmission and storage while still allowing additivity in the encrypted domain. EaSTFLy [69] builds upon the ternary gradients FL approach (TernGrad) [70], which quantizes the gradients into ternary vectors. The authors introduce a specialized batch encoding algorithm tailored for ternary gradients, along with a pair of encode-decode algorithms designed to prevent overflow issues. FHE-DiNN [71] implements BNN [72] on TFHE and TAPAS [73] also uses TFHE over BNN.

Discussion: We learned that quantization has two limitations. First, dequantization requires knowledge of the number of aggregated values, which poses challenges in flexible synchronization scenarios where the number of updates may vary or even be unknown. On the other hand, there are overflow issues in the aggregation phase. Since values are quantized into positive integers, aggregating them may lead to overflow as the sum grows larger. This requires the decryption of batched ciphertexts after a few additions and reencryption, which can cause inefficiencies [28].

2) Packing: In packing, which is also known as batching [63], the multiple local gradients are packed into a single plaintext message, helping to reduce the communication costs of the FL process while preserving privacy. Note that each client individually packs their own set of local gradients into a single plaintext message before encryption. For instance, Phong et al. [28] presented a pioneering framework for HFL

that leverages additive HE to safeguard the gradients against an honest-but-curious server. Next, in [28], it encodes the gradients of signed real numbers to nonnegative integers and then packs the nonnegative integers into a Paillier plaintext due to the huge bits in the plaintext space. In addition, they use zero padding to prevent overflows in ciphertext additions. Secure model fusion [74] is another optimization strategy by packing a group of weights into one operation that combines quantization and zero-padding techniques to avoid overflow to enhance computational efficiency. In addition, there are several novel packing techniques, i.e., Chinese remainder theorem (CRT) packing and polynomial packing, that can be utilized in HE-based PPFL. Zhang et al. [75] proposed to use CRT packing to pack the gradients and then use Paillier HE to encrypt the packed gradients. It also achieves verifiability by using a bilinear aggregate signature, which means it can detect malicious behavior, such as aggregation servers falsifying the aggregated gradient. VF2Boost [76] introduced a polynomial-based packing method that packs multiple feature histograms into a single ciphertext, leveraging the fact that the encoded integer of each value falls within a narrow range. ESAFL [77] employed CKKS' encoding method [30] to encode gradients into polynomials and designed the polynomial packing method to pack multiple local gradients into a single plaintext. Furthermore, they utilize the fast Fourier transform (FFT) for fast polynomial multiplication in plaintext encryption.

Discussion: In plaintext FL models, the server can receive the real value of gradients, and thus can use suitable bits to encode the sum or average to prevent overflows. However, in PPFL scenarios, the server does not have access to the real values as the gradients are either secret-shared or encrypted. In such cases, alternative solutions must be employed to address the overflow problem.

3) Transmitted Parameter Selection: TPS enables each client to only transmit a selected subset of its parameters for updating while the remaining parameters are accumulated and incorporated into future training iterations. Li et al. [78] introduced a novel double-end sparsification technique to diminish the transmission of additional intermediate results. In this method, both active and passive parties only transmit the results with the most significant changes, while other changes in the untransmitted results are accumulated locally until they reach a sufficient magnitude. Inspired by this approach, Yang et al. [79] utilized a similar sparsification strategy to optimize the HE to transfer parameters process in PPFL. Their methodology entails selectively transmits gradient updates exceeding a predetermined threshold, and concurrently accumulating updates that fall short of the threshold locally. In FLZip [80], instead of encrypting individual gradients, each client first filters insignificant gradients by considering the magnitude of the gradients in each layer independently. To allow aggregation to be performed on ciphertexts of the sparsified gradients, FLZip uses a key-value pair encoding scheme. Moreover, to counter the accuracy loss as a result of sparsification, FLZip also utilizes an error accumulation mechanism. Furthermore, AVFL [81] leveraged the principal component analysis (PCA) to perform dynamic feature selection, while CE-VFL [82] utilizes both PCA and autoencoders to learn latent representations from raw data. Another group of works is to disclose the partial local parameter for better efficiency. Previous work on privacy leakage analysis shows that the *partial transparency*, e.g., hiding parts of the models [83], [84], [85], can grant the adversary a limited chance to successfully perform attacks like gradient inversion privacy attacks [15]. FLARE [35] provided an available configuration to perform HE operations on certain layers. Furthermore, FedML [86] supported parameter-wise selection for more fine-grained overhead control with the corresponding privacy leakage analysis.

Discussion: It was learned that several limitations arise when integrating sparsification techniques with cryptographic techniques such as MPC or HE in PPFL. One challenge is to implement downlink sparsification (i.e., sparsifying the global update sent from the server to users) when the server is not aware of the plaintext values of the aggregated update due to cryptographic techniques.

B. Local Training Phase

1) Gradient Descent Optimization: GDO is another technique to save communication costs by reducing the number of communication rounds in FL while maintaining the performance of the aggregated model. To address this issue, some works optimize the gradient descent algorithm to reduce communication overhead in HE-based PPFL. Liu et al. [87] proposed a federated stochastic block coordinate descent algorithm (FeDBCD), which allows each party to conduct multiple client updates before each communication to reduce the number of synchronizations. They carefully select the appropriate local iteration numbers to balance the communication and computational overhead caused by HE operation to improve overall efficiency. Similarly, Wei et al. [88] increased local training rounds prior to gradient update to reduce the cost of information exchange and HE between both parties during asynchronous gradient sharing, improving training efficiency, and the corresponding speedup can be more than 10x compared to FedAvg. Zhang and Zhu [89] proposed a new HE-based backpropagation algorithm that preserves privacy, which is computationally efficient, and supports collaborative asymmetric machine learning. A deep neural network with a partially encrypted strategy is proposed for this scheme to avoid learning on encrypted data directly and can be more than $100\times$ times speedup compared to Cryptonets [90]. Yang et al. [91] adopted the quasi-Newton method for fast convergence, which is faster than first-order gradient-based methods [61]. VPFL [92] combined the distributed selective stochastic gradient descent (DSSGD) method with the Paillier homomorphic cryptosystem to achieve distributed encryption functionality in order to reduce the computational cost of the complex cryptosystem.

Discussion: It was observed that how to reasonably set the number of communication rounds is promising. The trade-off between computation and communication needs further investigation.

C. Server Aggregation Phase

1) Lightweight Homomorphic Encryption: Some of the work adopts lightweight HE approaches to reduce the overhead of HE. ACCEL [93] employed symmetric HE (SHE), which not only facilitates various homomorphic operations, but also provides an efficient and secure approach for the preparation, distribution, and computation of ciphertexts. In ACCEL, SHE is utilized to devise a data aggregation matrix construction protocol for vertical federated logistic regression, ultimately enhancing the training efficiency. Hao et al. [94] presented an efficient and secure gradient aggregation scheme in FL that exploits the lightweight symmetric AHE called PPDM [95], which is a newly devised technique for homomorphic data aggregation. It simultaneously supports aggregation of additions and multiplications with a unified mechanism from individual data in the encrypted domain, requiring it to perform any one-way and one-time trapdoor function computation. In addition, to further improve security and tolerate user dropouts, a DP technique is also utilized to add calibrated noise to each local gradient before encryption.

Discussion: It was found that lightweight HE is a good choice for cross-device FL since IoT devices are resource-constrained. However, this approach may potentially compromise security levels since symmetric encryption does not provide the same level of security as public-key encryption. Furthermore, key management is cumbersome in symmetric encryption, which limits the application of symmetric encryption in PPFL.

D. Overall FL Phases

1) End-to-End Optimization: End-to-end HE-based PPFL [96] refers to employing FHE techniques to achieve zero-leakage training of ML models in a federated setting where the client's local data, intermediate updates, and the final model remain encrypted. SPINDLE [97] is the first practical and efficient federated system that enables privacy-preserving training of a complete logistic-regression workflow using FHE. To enhance the efficiency of FHE in federated training, they propose several end-to-end optimizations, including parallel computations, SIMD operations, efficient collective operations, and optimized polynomial approximations for activation functions such as sigmoid and softmax. POSEIDON [98] follow-up work is the first end-to-end system that enables distributed learning on neural networks with multiple clients in an FL setting. POSEIDON employs several packing schemes to enable SIMD operations on the weights of various network layers and uses approximations that enable the evaluation of multiple activation functions (e.g., Sigmoid, Softmax, ReLU) under encryption. In addition, CryptoBoost [96] employs FHE to build XGBoost end-to-end secure federated and achieves more optimized system efficiency through new secure computation protocols such as secure division, secure comparison, and secure sorting.

Discussion: It was learned that end-to-end PPFL ensures stronger security and privacy. However, achieving end-to-end PPFL workflow requires the adoption of FHE, which may lead to impractical overheads for large-scale ML models. Thanks to the properties of multikey HE (MHE) [60], the mentioned work is also secure against $N-1$ colluding clients out of N clients.

2) Concurrent Training: VF2Boost [76] improved vertical federated GBDT by concurrent training to reduce idle waiting. In more detail, for the root node, a blaster-style encryption scheme is introduced to parallelize the encryption, public network communication, and histogram construction phases. For subsequent layers of the decision tree, an optimistic node-splitting strategy is developed to overlap the decryption and histogram construction phases. Moreover, Secureboost+ [66] reduces cipher-related computation costs and communication costs at a training mechanism level by using mix tree training and layered tree training.

IV. Hardware Acceleration

The design of hardware-based acceleration has been becoming more and more critical in FL because the adopted privacy-preserving methods could result in significant computation and communication costs. Existing research utilizes different hardware infrastructures, including field-programmable gate arrays (FPGAs) [101], [108], graphics processing unit (GPU) [99], and CPU [103] to achieve acceleration in HE-based PPFL.

A. FPGA-Based

FPGAs are semiconductor-integrated circuits [109], [110] that can be configured for various tasks after being manufactured. This adaptability enables FPGAs to speed up significant workloads and allows designers to adjust to evolving demands. FPGAs have been employed in diverse fields, including traditional machine learning [111] and cryptography [112]. In recent years, FPGAs have been utilized to boost the performance of VFL with HE. Yang et al. [101] introduced an FPGA-based HE framework to accelerate the training phase. They strategically offloaded the modular multiplication operation, the central component of the Paillier cryptosystem, onto the FPGAs. Moreover, they developed a streamlined architecture for the Paillier cryptosystem to integrate the FPGA framework into VFL. Additionally, FLASH [102] presented a more refined design for VFL based on the Paillier cryptosystem, enabling a broader range of cryptographic operations. Using FPGAs to speed up VFL processes allows researchers to improve computational efficiency and flexibly accommodate various requirements.

Discussion: FPGA provides the capability to design a hardware architecture specifically for efficient cross-silo FL. This is achieved by creating hardware circuits from the ground up, thereby enabling us to devise an optimized, fine-grained pipelining system with adaptable bit-width support, which expedites HE operations. Moreover FPGA furnishes ample on-chip memory, facilitating the storage of large numbers utilized in the processing pipeline.

B. GPU-Based

The GPU is available hardware for accelerating the training process of machine learning models due to its parallel architecture. In HE-based VFL, Cheng et al. [99] proposed

a GPU-based acceleration solution called HAFLO for the vertical federated logistic regression algorithm. HAFLO optimizes the core homomorphic operation to reduce the overhead introduced by the Paillier cryptosystem. Furthermore, HAFLO also optimizes IO and storage on GPU for the vertical federated logistic regression algorithm. Furthermore, for end-to-end HE-based PPFL [96], [97], [98], acceleration on the GPU could refer to [113] and [114]. Jung et al. [113] identified the main-memory bandwidth bottleneck as a key challenge in accelerating FHE operations using GPUs and developed a GPU implementation that extensively utilizes memory-centric optimizations. By applying their GPU implementation to train a logistic regression model, they achieved a significant speedup of 40× compared to an 8-threaded CPU implementation [115], [116]. CARM [100] is the first optimized GPU implementation of FHE schemes designed specifically for IoT scenarios. Although the mentioned work has not been implemented in the FL setting, they offer a promising solution for End-to-End HE-based PPFL.

Discussion: It was learned that GPU [117] is ideal for performing data parallelism over tensors with short numbers (e.g., single precision floats); however, it fails to provide efficient pipeline execution for HE operations with large numbers (e.g., 2048-bit integers).

C. CPU-Based

In contrast to earlier works that depend on specialized hardware, such as GPUs and FPGAs, Intel introduced the Intel HE acceleration library (HEXL) in 2021 [118]. HEXL takes advantage of the SIMD features and Intel AVX-512 instructions available on Intel CPUs, which are easily accessible, to deliver plug-and-play acceleration for FHE. For example, in [103], the modular exponentiation operation of partial HE in FL has been significantly enhanced by the introduction of the multibuffer function provided by HEXL.

Discussion: In this survey, we identified the main cause behind the inefficiency of PHE and FHE. PHE operations are based on two basic operators: modular exponentiation and modular multiplication. Most FHE operations are built upon polynomials, which are much more complex than plaintext computation, while FFT/number theoretic transform (NTT) can be utilized to speed up the polynomial operations from algorithm level, further accelerating NTT / FFT faces challenges in three aspects: high computation complexity, extremely intensive memory access, and limited generality.

V. Hybrid Optimization With Other Privacy-Preserving Techniques

Privacy-preserving techniques offer unique advantages when integrated with HE. Therefore, researchers adopt hybrid approaches to improve system efficiency and provide strong privacy guarantees in HE-based PPFL.

A. HE+DP

Kim et al. [104] proposed a new method to use a combination of HE and DP for an iterative learning algorithm in FL. They decrypted the model at each iteration, and the decrypted model is redistributed to generate fresh inputs for the next iteration. It focuses on solely using smaller HE parameters than the ones used in previous approaches [115], [116]. Consequently, it leads to better performance in terms of computational efficiency and accuracy. Chai et al. [105] proposed an efficient federated matrix factorization method that protects users against inference attacks in FL. The key idea is that they obfuscate one user's rating to another such that the private attribute leakage is minimized under the given distortion budget, which limits recommending loss and overhead of system efficiency. During the obfuscation, they apply DP to control information leakage between users. They also adopt HE to protect intermediate results during FL training.

B. HE+SS

Several works combine HE with SS to protect the intermediate computational results from leaking to any other client. For instance, Pivot [106] combines SS and HE to guarantee that no intermediate results are disclosed when aggregating general tree models, including RF and GBDT. In Pivot, HE is used extensively to facilitate local computations on the client side. SS is only invoked in the scenarios where HE proves to be inadequate in terms of functionality. By adopting this approach, the proposed solution can not only ensure a high level of security but also exhibit a remarkable improvement in efficiency when applied to the vertical tree models. PrivFL [34] enabled a robust and secure training process by iteratively executing a secure multiparty global gradient protocol and using lightweight HE operations, which are suitable for mobile applications. Specifically, PriVFL designed two privacy-preserving protocols for training linear and logistic regression models based on an additive HE and a masking protocol. CAESAR [107] combined HE and SS to build a secure large-scale sparse logistic regression model and achieve both efficiency and security. Moreover, they adopted mini-batch gradient descent, which benefits from fast convergence and enjoys good computation speed using vectorization libraries. FEDXGB [119] involves a new HE-based secure aggregation scheme for FL. By combining the advantages of SS and HE, the algorithm can force the server to conduct the aggregation operation and is robust against user dropout. An additive SS scheme is implemented in [120] to safeguard against collusion additionally, a discrete logarithm-based verification scheme is introduced, effectively ensuring result accuracy and identifying nonperforming servers with over 50% less overhead than traditional bilinear signature methods.

Discussion: The standard HE-based PPFL assumes that all clients are semi-honest and that there is no collusion between clients and servers. Thanks to SS, the mentioned work can resist collusion attacks from untrusted clients. We also observed that different security primitives have their pros and cons. For example, SS-based protocols often have large communication overheads but maintain model accuracy, whereas DP-based protocols can be more efficient, but may suffer from significant accuracy loss. In the future, we expect that researchers will continue to develop and explore hybrid

protocols to balance the tradeoffs between communication overhead, computational complexity, privacy guarantees, and model accuracy.

VI. Future Direction and Conclusion

It was found that the utilization of FHE in PPFL is a relatively unexplored area. There is a significant need for end-to-end privacy and robustness in FL, both of which require complex computations. Consequently, these requirements can only be feasibly achieved through the implementation of FHE. It was noticed that the current applications of FHE-based end-to-end encryption are considerably slow and not yet practical for widespread use. To date, there have been no attempts to integrate FHE-based robustness [121] into the context of FL. Therefore, it is necessary to optimize FHE computations and design a robust FL solution that leverages FHE. Furthermore, another promising direction for research could be the dynamic adjustment of privacy requirements based on different scenarios, applications, and data [5], [122]. Such an approach could effectively balance privacy and efficiency, thus achieving an optimal tradeoff.

In summary, this article has presented a comprehensive survey on the efficiency optimization of PPFL from the HE perspective. First, we have given an overview of FL and HE, including concepts, categories, their integration with HE-based PPFL, and the threat model. Then, we have introduced the three optimization skills from algorithmic, hardware, and hybrid perspectives and discuss their limitations and adaptions in HE-based PPFL. Finally, we have outlined existing challenges as well as several directions for future research.

Acknowledgment

The authors thank Dr Ziyao Liu and Xiaojun Xie for their effort on the early stage of this work.

References

[1] W. Ding, S. Jiang, H.-W. Chen, and M.-S. Chen, "Incremental reinforcement learning with dual-adaptive ε-greedy exploration," in *Proc. AAAI*, 2023, pp. 7387–7395.

[2] C. Wang, T. Hu, and S. Jiang, "Pairwise learning problems with regularization networks and Nyström subsampling approach," *Neural Netw.*, vol. 157, pp. 176–192, Jan. 2023.

[3] Z. Lian, W. Wang, Z. Han, and C. Su, "Blockchain-based personalized federated learning for Internet of Medical Things," *IEEE Trans. Sustain. Comput.*, vol. 8, no. 4, pp. 694–702, Dec. 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:258879366

[4] Z. Lian, Q. Zeng, W. Wang, T. R. Gadekallu, and C. Su, "Blockchain-based two-stage federated learning with non-IID data in IoMT system," *IEEE Trans. Comput. Soc. Syst.*, vol. 10, no. 4, pp. 1701–1710, Aug. 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:253791295

[5] Z. Zhao, K. Wang, N. Ling, and G. Xing, "EdgeML: An autoML framework for real-time deep learning on the edge," in *Proc. IOTDI*, 2021, pp. 133–144.

[6] Z. Zhao, Z. Jiang, N. Ling, X. Shuai, and G. Xing, "ECRT: An edge computing system for real-time image-based object tracking," in *Proc. 16th ACM Conf. Embed. Netw. Sensor Syst.*, New York, NY, USA, 2018, pp. 394–395. [Online]. Available: https://doi.org/10.1145/3274783.3275199

[7] X. Shuai, Y. Shen, S. Jiang, Z. Zhao, Z. Yan, and G. Xing, "BalanceFL: Addressing class imbalance in long-tail federated learning," in *Proc. IPSN*, 2022, pp. 271–284.

[8] B. Fan, S. Jiang, X. Su, and P. Hui, "Model-heterogeneous federated learning for Internet of Things: Enabling technologies and future directions," 2023, *arXiv:2312.12091*.

[9] S. Jiang, W. Ding, H.-W. Chen, and M.-S. Chen, "PGADA: Perturbation-guided adversarial alignment for few-shot learning under the support-query shift," in *Proc. PAKDD*, 2022, pp. 3–15.

[10] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[11] H.-Y. Chen and W.-L. Chao, "FedBE: Making Bayesian model ensemble applicable to federated learning," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–21.

[12] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *Proc. IEEE Symp. Security Privacy (SP)*, 2019, pp. 739–753.

[13] S. Truex, L. Liu, M. E. Gursoy, L. Yu, and W. Wei, "Demystifying membership inference attacks in machine learning as a service," *IEEE Trans. Services Comput.*, vol. 14, no. 6, pp. 2073–2089, Dec. 2021.

[14] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, "Beyond inferring class representatives: User-level privacy leakage from federated learning," in *Proc. IEEE INFOCOM IEEE Conf. Comput. Commun.*, 2019, pp. 2512–2520.

[15] B. Zhao, K. R. Mopuri, and H. Bilen, "iDLG: Improved deep leakage from gradients," 2020, *arXiv:2001.02610*.

[16] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the GAN: Information leakage from collaborative deep learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, pp. 603–618.

[17] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients-how easy is it to break privacy in federated learning?" in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 16937–16947.

[18] L. Lyu and C. Chen, "A novel attribute reconstruction attack in federated learning," 2021, *arXiv:2108.06910*.

[19] K. Bonawitz et al., "Practical secure aggregation for privacy-preserving machine learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, pp. 1175–1191.

[20] J. So et al., "Lightsecagg: A lightweight and versatile design for secure aggregation in federated learning," in *Proc. Mach. Learn. Syst.*, 2022, pp. 694–720.

[21] K. Wei et al., "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3454–3469, 2020.

[22] Z. Lian, W. Wang, and C. Su, "COFEL: Communication-efficient and optimized federated learning with local differential privacy," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2021, pp. 1–6. [Online]. Available: https://api.semanticscholar.org/CorpusID:236939817

[23] A. C.-C. Yao, "How to generate and exchange secrets," in *Proc. 27th Annu. Symp. Found. Comput. Sci. (SFCS)*, 1986, pp. 162–167.

[24] A. Beimel, "Secret-sharing schemes: A survey," in *Proc. Int. Conf. Coding Cryptol.*, 2011, pp. 11–46.

[25] J. Song, W. Wang, T. R. Gadekallu, J. Cao, and Y. Liu, "EPPDA: An efficient privacy-preserving data aggregation federated learning scheme," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 5, pp. 3047–3057, Oct. 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:247134378

[26] P. Sun, X. Chen, G. Liao, and J. Huang, "A profit-maximizing model marketplace with differentially private federated learning," in *Proc. IEEE Conf. Comput. Commun.*, 2022, pp. 1439–1448.

[27] B. Jayaraman and D. E. Evans, "Evaluating differentially private machine learning in practice," in *Proc. USENIX Secur. Symp.*, 2019, pp. 1–19.

[28] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Trans. Inf. Forensics Security*, vol. 13, pp. 1333–1345, 2017.

[29] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proc. 41st Annu. ACM Symp. Theory Comput.*, 2009, pp. 169–178.

[30] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *Proc. 23rd Int. Conf. Theory Appl. Cryptol. Inf. Secur., ASIACRYPT*, Hong Kong, China, 2017, pp. 409–437.

[31] F. Tang et al., "Solving small exponential ECDLP in EC-based additively homomorphic encryption and applications," Cryptol. ePrint Arch., IACR, Bellevue, WA, USA, Rep. 2022/1573, 2022. [Online]. Available: https://eprint.iacr.org/2022/1573

[32] H. Yang, S. Shen, S. Jiang, L. Zhou, W. Dai, and Y. Zhao, "XNET: A real-time unified secure inference framework using homomorphic encryption," Cryptol. ePrint Arch., IACR, Bellevue, WA, USA, Rep. 2023/1428, 2023.

[33] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Rev.*, vol. 41, no. 2, pp. 303–332, 1999.

[34] K. Mandal and G. Gong, "PrivFL: Practical privacy-preserving federated regressions on high-dimensional data over mobile networks," in *Proc. ACM SIGSAC Conf. Cloud Comput. Secur. Workshop*, 2019, pp. 57–68.

[35] H. R. Roth et al., "NVIDIA FLARE: Federated learning from simulation to real-world," 2022, *arXiv:2210.13291*.

[36] "FATE framework from webank," Webank. Accessed: Nov. 15, 2019. [Online]. Available: https://fate.fedai.org.

[37] H. Ludwig et al., "IBM federated learning: An enterprise framework white paper V 0.1," 2020, *arXiv:2007.10987*.

[38] D. Stripelis et al., "Secure neuroimaging analysis using federated learning with homomorphic encryption," in *Proc. 17th Int. Symp. Med. Inf. Process. Anal.*, 2021, pp. 351–359.

[39] J. Zhou et al., "Personalized and privacy-preserving federated heterogeneous medical image analysis with PPPML-HMI," medRxiv, 2023.

[40] L. Lyu et al., "Privacy and robustness in federated learning: Attacks and defenses," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Nov. 10, 2022, doi: 10.1109/TNNLS.2022.3216981.

[41] X. Yin, Y. Zhu, and J. Hu, "A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions," *ACM Comput. Surveys*, vol. 54, no. 6, pp. 1–36, 2021.

[42] Y. Liu et al., "Vertical federated learning," 2022, *arXiv:2211.12814*.

[43] L. Yang et al., "A survey on vertical federated learning: From a layered perspective," 2023, *arXiv:2304.01829*.

[44] M. Mansouri, M. Onen, W. B. Jaballah, and M. Conti, "SoK: Secure aggregation based on cryptographic schemes for federated learning," in *Proc. Privacy Enhanc. Technol*, 2023, pp. 140–157.

[45] Z. Liu, J. Guo, W. Yang, J. Fan, K.-Y. Lam, and J. Zhao, "Privacy-preserving aggregation in federated learning: A survey," *IEEE Trans. Big Data*, early access, Jul. 15, 2022, doi: 10.1109/TBDATA.2022.3190835.

[46] Z. Jiang, W. Wang, B. Li, and Q. Yang, "Towards efficient synchronous federated training: A survey on system optimization strategies," *IEEE Trans. Big Data*, vol. 9, no. 2, pp. 437–454, Apr. 2023.

[47] C. Juvekar, V. Vaikuntanathan, and A. Chandrakasan, "GAZELLE: A low latency framework for secure neural network inference," in *Proc. 27th USENIX Secur. Symp. (USENIX Secur.)*, 2018, pp. 1651–1669.

[48] F. Boemer, A. Costache, R. Cammarota, and C. Wierzynski, "nGraph-HE2: A high-throughput framework for neural network inference on encrypted data," in *Proc. 7th ACM Workshop Encrypt. Comput. Appl. Homomorph. Cryptogr.*, 2019, pp. 45–56.

[49] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–19, 2019.

[50] J. Lee, J. Sun, F. Wang, S. Wang, C.-H. Jun, and X. Jiang, "Privacy-preserving patient similarity learning in a federated environment: Development and analysis," *JMIR Med. Inform.*, vol. 6, no. 2, p. e20, 2018.

[51] B. Pfitzner, N. Steckhan, and B. Arnrich, "Federated learning in a medical context: A systematic literature review," *ACM Trans. Internet Techn.*, vol. 21, no. 2, pp. 1–31, 2021.

[52] P. Kairouz et al., "Advances and open problems in federated learning," *Found. Trends® Mach. Learn.*, vol. 14, nos. 1–2, pp. 1–210, 2021.

[53] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. Int. Conf. Theory Appl. Cryptogr. Techn.*, 1999, pp. 223–238.

[54] Z. Brakerski and V. Vaikuntanathan, "Fully homomorphic encryption from ring-LWE and security for key dependent messages," in *Proc. Annu. Cryptol. Conf.*, Santa Barbara, CA, USA, 2011, pp. 505–524.

[55] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) fully homomorphic encryption without bootstrapping," *ACM Trans. Comput. Theory (TOCT)*, vol. 6, no. 3, pp. 1–36, 2014.

[56] J. Fan and F. Vercauteren, "Somewhat practical fully homomorphic encryption," Cryptol. ePrint Arch., IACR, Bellevue, WA, USA, Rep. 2012/144, 2012.

[57] L. Ducas and D. Micciancio, "FHEW: Bootstrapping homomorphic encryption in less than a second," in *Proc. Annu. Int. Conf. Theory Appl. Cryptogr. Techn.*, 2015, pp. 617–640.

[58] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène, "TFHE: Fast fully homomorphic encryption over the torus," *J. Cryptology*, vol. 33, pp. 34–91, Apr. 2019.

[59] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, 2015, pp. 1310–1321.

[60] J. Ma, S.-A. Naas, S. Sigg, and X. Lyu, "Privacy-preserving federated learning based on multi-key homomorphic encryption," *Int. J. Intell. Syst.*, vol. 37, no. 9, pp. 5880–5901, 2022.

[61] S. Hardy et al., "Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption," 2017, *arXiv:1711.10677*.

[62] K. Cheng et al., "SecureBoost: A lossless federated learning framework," *IEEE Intell. Syst.*, vol. 36, no. 6, pp. 87–98, Dec. 2021.

[63] C. Zhang, S. Li, J. Xia, W. Wang, F. Yan, and Y. Liu, "BatchCrypt: Efficient homomorphic encryption for cross-silo federated learning," in *Proc. USENIX Annu. Tech. Conf. (USENIX ATC)*, 2020, pp. 493–506.

[64] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends® Theor. Comput. Sci.*, vol. 9, nos. 3–4, pp. 211–407, 2014.

[65] D. Evans, V. Kolesnikov, and M. Rosulek, "A pragmatic introduction to secure multi-party computation," *Found. Trends® Privacy Secur.*, vol. 2, nos. 2–3, pp. 70–246, 2018.

[66] W. Chen, G. Ma, T. Fan, Y. Kang, Q. Xu, and Q. Yang, "SecureBoost+: A high performance gradient boosting tree framework for large scale vertical federated learning," 2021, *arXiv:2110.10927*.

[67] W. Xu, H. Fan, K. Li, and K. Yang, "Efficient batch homomorphic encryption for vertically federated XGBoost," 2021, *arXiv:2112.04261*.

[68] J. Han and L. Yan, "Adaptive batch homomorphic encryption for joint federated learning in cross-device scenarios," *IEEE Internet Things J.*, vol. 11, no. 6, pp. 9338–9354, Mar. 2024.

[69] Y. Dong, X. Chen, L. Shen, and D. Wang, "EaSTFLy: Efficient and secure ternary federated learning," *Comput. Secur.*, vol. 94, Jul. 2020, Art. no. 101824.

[70] W. Wen et al., "Terngrad: Ternary gradients to reduce communication in distributed deep learning," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1508–1518.

[71] F. Bourse, M. Minelli, M. Minihold, and P. Paillier, "Fast homomorphic evaluation of deep discretized neural networks," in *Proc. 38th Annu. Int. Cryptol. Conf., Cryptol. CRYPTO*, Santa Barbara, CA, USA, 2018, pp. 483–512.

[72] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks," in *Proc. 30th Conf. Neural Inf. Process. Syst.*, 2016, pp. 1–9.

[73] A. Sanyal, M. Kusner, A. Gascon, and V. Kanade, "TAPAS: Tricks to accelerate (encrypted) prediction as a service," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4490–4499.

[74] C. Liu, S. Chakraborty, and D. Verma, "Secure model fusion for distributed learning using partial homomorphic encryption," in *Policy-Based Autonomic Data Governance*. Cham, Switzerland: Springer, 2019, pp. 154–179.

[75] X. Zhang, A. Fu, H. Wang, C. Zhou, and Z. Chen, "A privacy-preserving and verifiable federated learning scheme," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2020, pp. 1–6.

[76] F. Fu et al., "Vf2boost: Very fast vertical federated gradient boosting for cross-enterprise learning," in *Proc. Int. Conf. Manag. Data*, 2021, pp. 563–576.

[77] J. Wu, W. Zhang, and F. Luo, "ESAFL: Efficient secure additively homomorphic encryption for cross-silo federated learning," 2023, *arXiv:2305.08599*.

[78] M. Li, Y. Chen, Y. Wang, and Y. Pan, "Efficient asynchronous vertical federated learning via gradient prediction and double-end sparse compression," in *Proc. 16th Int. Conf. Control, Autom., Robot. Vis. (ICARCV)*, 2020, pp. 291–296.

[79] K. Yang, Z. Song, Y. Zhang, Y. Zhou, X. Sun, and J. Wang, "Model optimization method based on vertical federated learning," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2021, pp. 1–5.

[80] X. Feng and H. Du, "FLZip: An efficient and privacy-preserving framework for cross-silo federated learning," in *Proc. IEEE Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) Cyber, Phys. Soc. Comput. (CPSCom) Smart Data (SmartData) Congr. Cybermat. (Cybermat.)*, 2021, pp. 209–216.

[81] D. Cai et al., "Accelerating vertical federated learning," 2022, *arXiv:2207.11456*.

[82] A. Khan, M. ten Thij, and A. Wilbik, "Communication-efficient vertical federated learning," *Algorithms*, vol. 15, no. 8, p. 273, 2022.

[83] F. Mo, A. Borovykh, M. Malekzadeh, H. Haddadi, and S. Demetriou, "Layer-wise characterization of latent information leakage in federated learning," 2020, *arXiv:2010.08762*.

[84] A. Hatamizadeh et al., "Gradvit: Gradient inversion of vision transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 10021–10030.

[85] Z. Lian, W. Wang, H. Huang, and C. Su, "Layer-based communication-efficient federated learning with privacy preservation," *IEICE Trans. Inf. Syst.*, vol. 105-D, pp. 256–263, Feb. 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:246469846

[86] W. Jin et al., "FedML-HE: An efficient homomorphic-encryption-based privacy-preserving federated learning system," 2023, *arXiv:2303.10837*.

[87] Y. Liu et al., "A communication efficient collaborative learning framework for distributed features," 2019, *arXiv:1912.11187*.

[88] Q. Wei, Q. Li, Z. Zhou, Z. Ge, and Y. Zhang, "Privacy-preserving two-parties logistic regression on vertically partitioned data using asynchronous gradient sharing," *Peer-to-Peer Netw. Appl.*, vol. 14, pp. 1379–1387, May 2021.

[89] Y. Zhang and H. Zhu, "Additively homomorphical encryption based deep neural network for asymmetrically collaborative machine learning," 2020, *arXiv:2007.06849*.

[90] 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 *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 201–210.

[91] K. Yang, T. Fan, T. Chen, Y. Shi, and Q. Yang, "A quasi-Newton method based vertical federated learning framework for logistic regression," 2019, *arXiv:1912.00513*.

[92] J. Zhang, Y. Liu, D. Wu, S. Lou, B. Chen, and S. Yu, "VPFL: A verifiable privacy-preserving federated learning scheme for edge computing systems," *Digit. Commun. Netw.*, vol. 9, no. 4, pp. 981–989, 2022.

[93] J. Zhao et al., "ACCEL: An efficient and privacy-preserving federated logistic regression scheme over vertically partitioned data," *Sci. China Inf. Sci.*, vol. 65, no. 7, 2022, Art. no. 170307.

[94] M. Hao, H. Li, G. Xu, S. Liu, and H. Yang, "Towards efficient and privacy-preserving federated deep learning," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2019, pp. 1–6.

[95] J. Zhou, Z. Cao, X. Dong, and X. Lin, "PPDM: A privacy-preserving protocol for cloud-assisted e-healthcare systems," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 7, pp. 1332–1344, Oct. 2015.

[96] C. Jin et al., "Towards end-to-end secure and efficient federated learning for XGBoost," in *Proc. AAAI*, 2022, pp. 1–9.

[97] D. Froelicher et al., "Scalable privacy-preserving distributed learning," 2020, *arXiv:2005.09532*.

[98] S. Sav et al., "Poseidon: Privacy-preserving federated neural network learning," 2020, *arXiv:2009.00349*.

[99] X. Cheng, W. Lu, X. Huang, S. Hu, and K. Chen, "HAFLO: GPU-based acceleration for federated logistic regression," 2021, *arXiv:2107.13797*.

[100] S. Ying Shen, H. Yang, Y. Liu, Z. Liu, and Y. Zhao, "CARM: CUDA-accelerated RNS multiplication in word-wise homomorphic encryption schemes for Internet of Things," *IEEE Trans. Comput.*, vol. 72, no. 7, pp. 1999–2010, Jul. 2023.

[101] Z. Yang, S. Hu, and K. Chen, "FPGA-based hardware accelerator of homomorphic encryption for efficient federated learning," 2020, *arXiv:2007.10560*.

[102] J. Zhang, X. Cheng, W. Wang, L. Yang, J. Hu, and K. Chen, "{FLASH}: Towards a high-performance hardware acceleration architecture for cross-silo federated learning," in *Proc. 20th USENIX Symp. Netw. Syst. Des. Implement. (NSDI)*, 2023, pp. 1057–1079.

[103] (WeBank Intel, Santa Clara, CA, USA). *Accelerating Secure Computing for Federated Learning*. (2021). [Online]. Available: https://www.intel.com/content/www/us/en/customer-spotlight/stories/webank-customer-story.html.

[104] M. Kim, J. Lee, L. Ohno-Machado, and X. Jiang, "Secure and differentially private logistic regression for horizontally distributed data," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 695–710, 2019.

[105] D. Chai, L. Wang, K. Chen, and Q. Yang, "Efficient federated matrix factorization against inference attacks," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 4, pp. 1–20, 2022.

[106] Y. Wu, S. Cai, X. Xiao, G. Chen, and B. C. Ooi, "Privacy preserving vertical federated learning for tree-based models," *Proc. VLDB Endowment*, vol. 13, no. 12, pp. 2090–2103, 2020.

[107] C. Chen et al., "When homomorphic encryption marries secret sharing: Secure large-scale sparse logistic regression and applications in risk control," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discovery Data Min.*, 2021, pp. 2652–2662.

[108] Z. Wang et al., "PipeFL: Hardware/software co-design of an FPGA accelerator for federated learning," *IEEE Access*, vol. 10, pp. 98649–98661, 2022.

[109] S. Jiang, H.-W. Chen, and M.-S. Chen, "Dataflow systolic array implementations of exploring dual-triangular structure in QR decomposition using high-level synthesis," in *Proc. ICFPT*, 2021, pp. 1–4.

[110] R. Fang, S. Jiang, H.-W. Chen, W. Ding, and M.-S. Chen, "Dual-triangular QR decomposition with global acceleration and partially Q-rotation skipping," in *Proc. ICFPT*, 2022, pp. 1–4.

[111] C. Zhang, P. Li, G. Sun, Y. Guan, B. Xiao, and J. Cong, "Optimizing FPGA-based accelerator design for deep convolutional neural networks," in *Proc. ACM/SIGDA Int. Symp. Field-Program. Gate Arrays*, 2015, pp. 161–170.

[112] M. S. Riazi, K. Laine, B. Pelton, and W. Dai, "HEAX: An architecture for computing on encrypted data," in *Proc. 25th Int. Conf. Architect. Support Program. Lang. Oper. Syst.*, 2019, pp. 1295–1309.

[113] W. Jung, S. Kim, J. H. Ahn, J. H. Cheon, and Y. Lee, "Over 100x faster bootstrapping in fully homomorphic encryption through memory-centric optimization with GPUs," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2021, no. 4, pp. 114–148, 2021.

[114] J. Zhang, X. Cheng, L. Yang, J. Hu, X. Liu, and K. Chen, "SoK: Fully homomorphic encryption accelerators," 2022, *arXiv:2212.01713*.

[115] M. Kim, Y. Song, S. Wang, Y. Xia, and X. Jiang, "Secure logistic regression based on homomorphic encryption: Design and evaluation," *JMIR Med. Inform.*, vol. 6, no. 2, 2018, Art. no. e8805.

[116] A. Kim, Y. Song, M. Kim, K. Lee, and J. H. Cheon, "Logistic regression model training based on the approximate homomorphic encryption," *BMC Med. Genom.*, vol. 11, no. 4, pp. 23–31, 2018.

[117] Z. Zhao, N. Ling, N. Guan, and G. Xing, "Aaron: Compile-time kernel adaptation for multi-DNN inference acceleration on edge GPU," in *Proc. 20th ACM Conf. Embed. Netw. Sensor Syst.*, 2022, pp. 802–803.

[118] F. Boemer, S. Kim, G. Seifu, F. D. M. de Souza, and V. Gopal, "Intel HEXL: Accelerating homomorphic encryption with Intel AVX512-IFMA52," in *Proc. 9th Workshop Encrypt. Comput. Appl. Homomorph. Cryptogr.*, 2021, pp. 57–62.

[119] Y. Liu et al., "Boosting privately: Federated extreme gradient boosting for mobile crowdsensing," in *Proc. IEEE 40th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2020, pp. 1–11.

[120] L. Lin and X. Zhang, "PPVerifier: A privacy-preserving and verifiable federated learning method in cloud-edge collaborative computing environment," *IEEE Internet Things J.*, vol. 10, no. 10, pp. 8878–8892, May 2022.

[121] Z. Ma, J. Ma, Y. Miao, Y. Li, and R. H. Deng, "ShieldFL: Mitigating model poisoning attacks in privacy-preserving federated learning," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 1639–1654, 2022.

[122] Z. Zhao, N. Ling, N. Guan, and G. Xing, "Miriam: Exploiting elastic kernels for real-time multi-DNN inference on edge GPU," 2023, *arXiv:2307.04339*.