Artificial Intelligence Review (2025) 58:82
https://doi.org/10.1007/s10462-024-11076-8

Approximate homomorphic encryption based privacy-preserving machine learning: a survey

Jiangjun Yuan[1] · Weinan Liu[1] · Jiawen Shi[1] · Qingqing Li[2]

Accepted: 13 December 2024 / Published online: 6 January 2025

Abstract
Machine Learning (ML) is rapidly advancing, enabling various applications that improve people's work and daily lives. However, this technical progress brings privacy concerns, leading to the emergence of Privacy-Preserving Machine Learning (PPML) as a popular research topic. In this work, we investigate the privacy protection topic in ML, and showcase the advantages of Homomorphic Encryption (HE) among different privacy-preserving techniques. Additionally, this work presents an introduction of approximate HE, emphasizing its advantages and providing the detail of some representative schemes. Moreover, we systematically review the related works about approximate HE based PPML schemes from the four technical applications and three advanced applications, along with their application scenarios, models and datasets. Finally, we suggest some potential future directions to guide readers in extending the research of PPML.

Keywords Privacy-preserving machine learning · Machine learning · Homomorphic encryption · Privacy

Jiangjun Yuan, Jiawen Shi and Qingqing Li have been contributed equally to this work.

✉ Weinan Liu
lwn@hzvtc.edu.cn

Jiangjun Yuan
yjj@hzvtc.edu.cn

Jiawen Shi
sjw@hzvtc.edu.cn

Qingqing Li
liqq@hzcu.edu.cn

[1] Business and Tourism Institute, Hangzhou Vocational & Technical College, Hangzhou 310018, Zhejiang, China

[2] Supercomputing Center, Hangzhou City University, Hangzhou 310000, Zhejiang, China

1 Introduction

With the rapid development of big data and Artificial Intelligence (AI) technology, Machine Learning (ML) has become an indispensable part of current work and daily lives. ML is an important branch in the field of AI that enables computers to learn from data and make decisions or predictions by building and training models (Michalski et al. 2013). In the past, traditional ML algorithms required hand-designed features and statistical information (Hsu and Lu 2023). However, the increase in computing power has made it possible to process large scale data, providing a strong impetus for the development of ML. Additionally, the rise of deep learning enables automatic feature extraction from raw data by neural networks, significantly improving the performance and effectiveness of ML models. The development of cloud and distributed computing technology further enables ML algorithms to run efficiently on distributed systems, accelerating the model training and inference process. Machine Learning as a Service (MLaaS) has accelerated its integration into people's work and daily lives (Vartiainen et al. 2020). For example, in the medical field, the application of ML offers new possibilities for disease diagnosis and treatment (Myszczynska et al. 2020). Through the analysis of medical records and case information from a large number of patients, ML can rapidly and accurately diagnose diseases and provide personalized treatment plans, which not only enhances the efficiency of doctors but also reduces the risk of misdiagnosis and underdiagnosis, leading to improved patient outcomes and survival rates.

However, the issue of privacy leakage is becoming increasingly prominent in the use of MLaaS (Rigaki and García 2023). It is important to consider the potential risks to users' personal privacy information during data collection, storage, processing, and transmission. The protection of users' personal information faces greater challenges, especially with the popularization of the Internet and mobile devices. Data is widely collected and used, making it difficult for users to control their private data. In 2018, a serious data breach occurred on Facebook, one of the world's largest social media platforms (Isaak and Hanna 2018). It was reported that the personal data of over 87 million users worldwide was illegally accessed and misused, including user profiles, friend lists, and personal preferences. The consequences of user data leakage may include identity theft, fraudulent activities and the exposure of personal information (Zou et al. 2020). To protect personal information, users should take necessary security measures when using the Internet services, especially MLaaS, while companies providing these services must manage user data in a legally compliant manner.

As privacy leakage becomes an increasingly pressing issue, users are placing greater emphasis on privacy protection. To strengthen the protection of personal data and regulate the collection and use of these data, many countries and regions have issued laws and regulations related to data and information security, like the California Consumer Privacy Act (Baik 2020) and the General Data Protection Regulation (Kok et al. 2023). Users desire to safeguard their privacy rights while enjoying the convenience of MLaaS, which renders pure ML services inadequate in meeting users' needs. As a result, Privacy-Preserving Machine Learning (PPML) services have emerged and are gradually becoming a new trend in MLaaS (Hesamifard et al. 2018; Wang et al. 2023). PPML services strengthen the privacy-preserving mechanism throughout the entire process of data collection, storage, processing, and transmission to prevent the leakage and abuse of users' private data. These services use encryption algorithms, privacy-preserving techniques, and data anonymization to achieve privacy and compliant use of user data, providing users with a more secure and reliable

MLaaS (Wood et al. 2020; Marcolla et al. 2022; Sousa and Kern 2023). The development of PPMLaaS can provide users with a better privacy-preserving experience, enhance user trust and promote a healthier and more sustainable industry of MLaaS.

1.1 Comparison with Existing Surveys

Table 1 presents several surveys relevant to this study. Given that our work focuses on ML based on approximate HE, the comparative surveys encompass three search fields: ML, HE, and PPML. The surveys listed in Table 1 are all obtained by searching Google Scholar. For the search of comparative literature in this part, we utilize the format "research field + survey or review". For example, to find review articles related to ML, we search using terms like "machine learning + survey" or "machine learning + review". Similarly, subsequent PPML schemes based on approximate HE are also obtained by searching Google Scholar using specific keywords. The keywords for this part include "CKKS (short for Cheon-Kim-Kim-Song) + technical application" or "approximate homomorphic encryption + technical application". For example, to compile literature on image processing, we utilize keyword combinations such as "CKKS + image processing" and "approximate homomorphic encryption + image processing".

Table 1 details the publication years of these surveys, as well as which of the three search fields they address. In the table, "/" indicates that the corresponding field is not covered, while "✓" signifies that it is included. For example, the work published in 1983 (Wood et al. 2020) addresses only ML and does not cover the other two search fields.

In the research field of ML, Carbonell et al. introduced ML from various perspectives (Carbonell et al. 1983). Jordan and Mitchell discussed the development of ML, its rich

Table 1 Comparison among Existing Surveys

Pub.	Year	ML involved	HE involved	PPML involved
Carbonell et al. (1983)	1983	✓	/	/
Aguilar-Melchor et al. (2013)	2013	/	✓	/
Jordan and Mitchell (2015)	2015	✓	/	/
Martins et al. (2017)	2017	/	✓	/
Acar et al. (2018)	2018	/	✓	/
Alaya et al. (2020)	2020	/	✓	/
Wood et al. (2020)	2020	✓	✓	✓
Liu et al. (2021)	2021	✓	✓	✓
Murshed et al. (2021)	2021	✓	/	✓
Munjal and Bhatia (2022)	2022	/	✓	/
Marcolla et al. (2022)	2022	✓	✓	✓
Rigaki and García (2023)	2023	✓	/	✓
Yao et al. (2023)	2023	✓	/	/
Sousa and Kern (2023)	2023	✓	✓	✓

application scenarios, and the modeling algorithms from multiple viewpoints, and they also provided an outlook on its future development (Jordan and Mitchell 2015). Yao et al. analyzed examples of how ML drives research in the energy sector, presented both current and future challenges, and described how ML can be effectively utilized for innovative research in this field (Yao et al. 2023).

In the research field of HE, Martins et al. introduced Fully Homomorphic Encryption (FHE) from an engineering perspective (Martins et al. 2017). Acar et al. introduced the theory and implementation of HE (Acar et al. 2018), and Alaya et al. introduced the development trend of HE schemes along with the related problems and challenges (Alaya et al. 2020). In addition, Aguilar-Melchor et al. presented some HE schemes and their applications in signal processing (Aguilar-Melchor et al. 2013), while Munjal and Bhatia presented the HE schemes in the health industry (Munjal and Bhatia 2022). Similarly, Marcolla et al. provided an overview on HE and its applications, but they focused on a wider range of application scenarios, including ML, cloud computing, and etc. (Marcolla et al. 2022).

There are a lot of works on PPML. For example, María and Sebastián focused on privacy attacks in ML and covered some protection techniques such as DP (Rigaki and García 2023). Murshed et al. reviewed the works on ML at the Network Edge, which also includes PPML as well as the topics of privacy and security (Murshed et al. 2021). Although these works explored PPML, they do not cover the application of HE in PPML. There are also related works that synthesize a variety of techniques used to protect the privacy of ML, including the work proposed by Liu et al., which introduces a variety of methods for protecting the privacy of ML, such as HE, Secure Multi-Party Computation (SMPC), DP, and Federated Learning (FL) (Liu et al. 2021).

There is a relatively small amount of review works on HE and PPML, which are the closest to the topic explored in our work. Wood et al. introduced HE and ML, and presented their applications in the medical and bioinformatics fields (Wood et al. 2020). In addition, Sousa and Kern focused on the field of Natural Language Processing (NLP) technology and analyzed the application of privacy-preserving deep learning in this field from three perspectives: data protection methods, trust methods, and authentication methods, where HE is only briefly introduced in the part of trust methods (Sousa and Kern 2023).

1.2 Motivation

Compared with other types of HE schemes, approximate HE schemes support floating-point computation, obtain approximate computation results, and provide Single Instruction Multiple Data (SIMD) to facilitate parallel and efficient processing of some computational operations. ML, on the other hand, requires floating-point computation, tolerates some loss of accuracy, and supports for various types of parallel algorithms. Approximate HE and ML complement each other in terms of functionality and requirements, and thus, approximate HE has been widely used in various types of PPML schemes, such as feature processing (Kim et al. 2018a; Lu et al. 2021; Koseki et al. 2023) and image processing (Jiang et al. 2018; Lee et al. 2022a; Kim and Guyot 2023).

By combing through Table 1, we find that there is no any review work specifically on the combination of approximate HE into ML. Therefore, we conduct an in-depth study on this topic, combing through the related literature and analyzing not only the NLP, i.e., text processing technical application, but also the technical applications of feature processing, image processing and audio processing, aiming to fill in the research gaps.

1.3 Contributions

The contributions of this work can be summarized as follows:

- We emphasize the significance and synergy between approximate HE and ML. This serves as a starting point for a literature review that enhances the research field of PPML.
- We review PPML schemes based on approximate HE across four technical applications: feature processing, text processing, image processing, and audio processing. Additionally, we provide a survey of three advanced technical applications: video processing, large language models (LLMs), and FL.
- We categorize the application scenarios of PPML schemes based on approximate HE, including commonly used models and datasets, as well as whether the scheme implementation is open source. This facilitates researchers entering this field to obtain comprehensive information and carry out related research more effectively.
- We further point out some future directions that aim to further deepen the research on approximate HE based PPML, enabling users to use ML services more securely.

1.4 Organization

The rest of this work is organized as follows: In Sect. 2, the background information is presented, including ML, HE and PPML. In Sect. 3, the approximate HE based PPML schemes are reviewed from four popular technical applications and three advanced applications. In Sect. 4, we make some discussions on the application scenarios/models/datasets of PPML based on approximate HE and some future directions. Finally, Sect. 5 makes the conclusion.

2 Background

In this section, we first introduce HE and PPML. For PPML, we list a number of open source projects, so that researchers can use them in the implementation of their own PPML schemes.

2.1 A Brief Introduction to HE

The concept of HE was first proposed in 1978 (Rivest et al. 1978), and in the same year, Rivest et al. implemented the RSA public key encryption algorithm (Rivest et al. 1978), which supports multiplicative homomorphism. Compared with multiplicative HE, additive HE is more widely used, e.g., data aggregation for smart grid (Zhao et al. 2023). The Benaloh encryption system proposed in 1994 implements additive HE (Benaloh 1994). To further extend the functionality of HE systems, the Paillier encryption system was designed in 1997 (Paillier 1999), which not only implements homomorphic addition operations, but also supports scalar multiplication operations, which makes the Paillier encryption system capable of realizing two different homomophic operations in a broad sense. Therefore, the

Paillier scheme is widely used in ML (He et al. 2022; Han and Yan 2023). For some application scenarios with higher security requirements, it is necessary to support both homomorphic addition operations and homomophic multiplication (not scalar multiplication) operations in a narrow sense. To fulfill these application scenarios, a number of novel HE schemes have been proposed, such as the BGN encryption system proposed in 2005 (Boneh et al. 2005). These schemes are usually able to support an infinite number of homomophic addition operations, but can only support a finite number of homomophic multiplication operations In 2009, Gentry's thesis pointed out a new path to realize FHE (Gentry 2009), which implements re-encryption by bootstrapping, making the modulus large enough to continue homomophic multiplication operations. However, Gentry's scheme is based on the difficult problems that incur significant computational overhead. With the in-depth research and development, FHE schemes that can be applied to practical applications have been proposed, represented by the BGV/BFV schemes (Brakerski et al. 2014; Fan and Vercauteren 2012; Brakerski and Vaikuntanathan 2014). These schemes support parallel and efficient SIMD operations, which make the amortized time shorter, greatly accelerating the computational efficiency.

In addition to the above encryption systems that support word-wise homomorphic addition and multiplication operations, there are also some HE schemes that support boolean circuit operations, i.e., bit-wise operations, and some representative schemes of third-generation FHE schemes, such as GSW (Gentry et al. 2013), FHEW (Ducas and Micciancio 2015), and TFHE (Chillotti et al. 2016). Different types of HE schemes have their own advantages. For example, the third generation FHE schemes are particularly fast in computation, but it does not support SIMD operations. While the BGV/BFV schemes, which support arithmetic circuits, are slow in computation on a single ciphertext, but it supports SIMD operations, and the amortized time can be greatly reduced. Currently, there are already some schemes that support the conversion between different types of HE schemes, such as PEGASUS (Lu et al. 2021).

Additionally, compared with some earlier HE schemes, such as RSA, the current HE schemes, like BGV, which are based on the hard problems on lattice, Learn With Error (LWE) (Regev 2009) or Ring Learn With Error (RLWE) (Lyubashevsky et al. 2010). These schemes belong to a new generation of cryptography with higher security and can withstand attacks from quantum computers.

Based on the BGV scheme, in 2017, Cheon et al. designed a novel HE scheme (Cheon et al. 2017), CKKS, which treats the error e caused in the whole encryption/decryption process as a part of the plaintext m, and does not remove e when decrypting. In addition, they devised an encoding mechanism that extends the numbers from integers to complex numbers. Previously, applying HE to ML required switching computation between integers and real numbers, making the computation overhead greatly increased. With the CKKS scheme, the facts that complex numbers naturally contain real numbers and that ML can tolerate a certain amount of error, make the CKKS scheme naturally suitable for ML. In addition to the ML domain discussed in this work, it also includes numerous application scenarios, such as statistical analysis (Jin et al. 2022; Chen et al. 2022) and image watermarking (Lai et al. 2022; Basuki et al. 2022).

The CKKS scheme requires an encoding operation on the plaintext to change an input vector in the complex domain $z \in \mathbb{C}^{N/2}$ to a polynomial $m(X) \in R$ before encryption, where N is the degree of the M-th cyclotomic polynomial $\Phi_M(X)$, and $M = 2N$. After decryption, the polynomial $m(X)$ needs to be decoded to change it back to the original vector z. Thus, encoding and decoding are two very important operations in the CKKS scheme. As an encryption and decryption system, the CKKS scheme also has key generation, encryption and decryption operations, as well as homomorphic addition and multiplication operations needed as an HE scheme. The CKKS scheme only supports a certain number of homomorphic multiplication operations, defined as L. There are a set of modulus q_l, denoted as $q_L, q_{L-1}, ..., q_1$. Since there is an extra scale factor after the multiplication operation, a rescaling operation is needed to offset it. Furthermore, the CKKS scheme also supports the key rotation operation to facilitate some operations, such as summation through rotation.

- KeyGen(1^λ): λ is the security parameter. For the three given generators:$\mathcal{HWT}$, U and DG, the output of the first one satisfies the discrete Gaussian distribution while the outputs of the latter two satisfy the uniform distribution, and the three parameters are generated: a, s, e, according to Eq. 1:

$$\begin{cases} s \leftarrow \mathcal{HWT}(h) \\ a \leftarrow U(R_{q_L}) \\ e \leftarrow DG(\sigma^2) \end{cases} \tag{1}$$

where h and R_{q_L} are two integers, and σ^2 is the variance of the DG generator. Generate samples from the three generators: $s \leftarrow \mathcal{HWT}(h)$, $a \leftarrow U(R_{q_L})$, and $e \leftarrow DG(\sigma^2)$, which are further used to generate two keys, the secret key $sk = s$, the public key $pk = (-a \cdot s + e, a) = (b, a) \in R^2_{P \cdot q_L}$.

Similarly, generate samples from the two generators: $a' \leftarrow U(P \cdot R_{q_L})$, and $e' \leftarrow DG(\sigma^2)$, and use them to generate the evaluate key $evk = (-a' \cdot s + e' + Ps^2 (\text{mod } P \cdot q_L), a') = (b', a')$. Above all, the private key, sk, the public key pk and the evaluate key evk can be generated as Eq. 2:

$$\begin{cases} sk = s \\ pk = (-a \cdot s + e, a) = (b, a) \\ evk = (-a' \cdot s + e' + Ps^2 (\text{mod } P \cdot q_L), a') = (b', a') \end{cases} \tag{2}$$

- Enc: Encrypt the polynomial generated from the encoding procedure m with pk, and obtain $c = (m + b, a) = (c_0, c_1) \in R^2_{q_l}$.

$$\begin{aligned} c &= u \cdot pk + (e_1, e_2) + (m, 0) \\ &= (ub + e_1 + m, ua + e_2) \\ &= (c_0, c_1) \end{aligned} \tag{3}$$

- Dec: The decryption format is $c_0 + c_1 \cdot s$, and we can derive an approximate m from Equation (4):

$$\langle c, sk \rangle = c \cdot (1, s) = ue + e_1 + e_2 s + m = m + e' \approx m \tag{4}$$

- Add & Mult: Assume two plaintexts are m and m'. Therefore, their ciphertexts are $\text{Enc}(m)$ and $\text{Enc}(m')$, respectively. Then, we have the homomorphic addition operation

$\mathrm{Enc}(m) + \mathrm{Enc}(m') = \mathrm{Enc}(m + m')$. Similarly, we have the homomorphic multiplication operation $\mathrm{Enc}(m) \cdot \mathrm{Enc}(m') = \mathrm{Enc}(m \cdot m')$. As the homomorphic multiplication operation introduces s^2 which should be removed by the relinearization operation with the evaluation key, *evk*.

- Rescaling: The scale factor Δ is used in the encoding and decoding processes, which can actually be viewed as $\mathrm{Enc}(\Delta m)$ and $\mathrm{Enc}(\Delta m')$ in the encryption process. When performing encrypted homomorphic operations, $\mathrm{Enc}(\Delta m) + \mathrm{Enc}(\Delta m') = \mathrm{Enc}(\Delta(m + m')$, and this scaling factor can be removed during the decoding process. However, when performing homomorphic multiplication operation, $\mathrm{Enc}(\Delta m) \cdot \mathrm{Enc}(\Delta m') = \mathrm{Enc}(\Delta^2(m \cdot m')$, there is one more scaling factor, and the amplification of the product becomes the square of the original one. Therefore, a rescaling procedure is required to remove this extra scaling factor, which is shown as Equation (5):

$$RS_{l \to l-1}(c) = \left\lfloor \frac{q_{l-1}}{q_l} c \right\rceil \pmod{q_{l-1}} \tag{5}$$

When applying the CKKS scheme to implement PPML schemes, it is generally necessary to set the security level to exceed 128, specifically $\lambda > 128$. And with a larger ring dimension *N* and a smaller ciphertext modulus *q*, a better security level of RLWE-based schemes can be achieved. For instance, to achieve 128-bit security, when *N* equals 8192, *q* must be less than 218 bits. When selecting secure parameters for approximate homomorphic encryption schemes, the lattice estimator (Albrecht et al. 2015) can serve as a highly useful tool.

Additionally, in a recent work Guo et al. (2024), Guo et al. presented application-specific attacks against CKKS. Therefore, when using open-source libraries to implement approximate HE based PPML schemes, careful attention must be paid to the setting of parameters, especially,the scaling factor Δ to achieve IND-CPA security. According to the work Cheon et al. (2024), to satisfy IND-CPA security, Lattigo requires a scaling factor Δ to be set within $2^{34} - 2^{51}$, while using OpenFHE (Al Badawi et al. 2022) and Simple Encrypted Arithmetic Library (SEAL) (Microsoft SEAL 2023), Δ should be set within $2^{48} - 2^{55}$.

For the three parameters mentioned above, *N*, *p*, and Δ, they need to be appropriately configured when using specific open-source libraries. Taking TenSEAL as an example, these three parameters correspond to: “poly_modulus_degree”, “coeff_mod_bit_sizes”, and “global_scale”, respectively. Here, “coeff_mod_bit_sizes” is a vector where, apart from the first and last elements, all intermediate elements should match Δ, i.e., “global_scale”. Setting the first two parameters to 8192 and [55, 50, 50, 55], respectively, ensures that the sum of the “coeff_mod_bit_sizes” array, which represents the modulus of the ciphertext, is less than 218, thereby meeting the requirement for 128-bit security. Additionally, setting “global_scale” to 50 will comply with the scaling factor Δ requirements for the SEAL open-source library as described in Cheon et al. (2024).

Over the years, such HE schemes based on the CKKS scheme have evolved considerably, deriving four main variants: Fully-CKKS which is the FHE version of the CKKS scheme, RNS-CKKS which is optimized by the Residual Number System (RNS), MK-CKKS which implements multi-key version, and torus-CKKS which can switch forth and back between LWE and RLWE ciphertext. There are also some novel schemes implementing multiple variants as shown in Fig. 1.

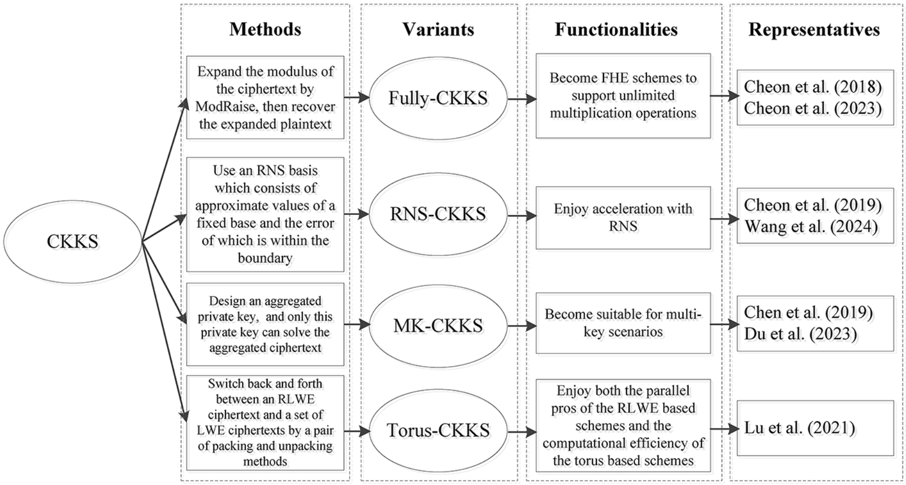
Methods
Variants
Functionalities
Representatives
CKKS
Expand the modulus of the ciphertext by ModRaise, then recover the expanded plaintext
Fully-CKKS
Become FHE schemes to support unlimited multiplication operations
Cheon et al. (2018)
Cheon et al. (2023)
Use an RNS basis which consists of approximate values of a fixed base and the error of which is within the boundary
RNS-CKKS
Enjoy acceleration with RNS
Cheon et al. (2019)
Wang et al. (2024)
Design an aggregated private key, and only this private key can solve the aggregated ciphertext
MK-CKKS
Become suitable for multi-key scenarios
Chen et al. (2019)
Du et al. (2023)
Switch back and forth between an RLWE ciphertext and a set of LWE ciphertexts by a pair of packing and unpacking methods
Torus-CKKS
Enjoy both the parallel pros of the RLWE based schemes and the computational efficiency of the torus based schemes
Lu et al. (2021)

Fig. 1 The representatives of approximate HE schemes

The work Cheon et al. (2018) is a representative of the Fully-CKKS variant. In the original CKKS scheme, when the defined level is used up, a bootstrapping process is required to recharge the level to the maximum one. The two important steps in the bootstrapping process are the ModRaise operation on the ciphertext and the EvalMod operation on the plaintext (Cheon et al. 2018). The ModRaise operation enlarges the modulus of the ciphertext which amplifies the plaintext from pt to $pt + qI$. The qI part should be removed and pt is recovered, which is implemented by the EvalMod operation. Recent advancements in enhancing the bootstrapping of approximate homomorphic encryption (HE) schemes can increase precision by decomposing a ciphertext into t components (Cheon et al. 2023). When t equals 2 (or 3), a doubling (or tripling) of precision can be attained.

The work Cheon et al. (2019) is a representative of the RNS-CKKS variant. In the CKKS scheme, the modulus on the ciphertext can not be decomposed into a set of co-primes. Therefore, it is difficult to integrate RNS. The work in Cheon et al. (2019) used an RNS basis which consists of approximate values of a fixed base and the error of which is within a defined boundary. For a given scale parameter q and a bit precision parameter η, the desired basis $C = \{q_0, ...q_L\}$ should meet the requirement of q/q_l. Then the new modulus on the ciphertext at the l-th level can be set as $Q_l = \prod_{i=0}^{l} q_i$, and the scaling ratio at the l-th level $q_l = Q_l/Q_{l-1}$, can be approximated to the scale factor q. Due to the correlation between RNS efficiency and hardware, recent work has focused on optimizing RNS-CKKS at the hardware level. For instance, in the study by Wang et al. (2024), a butterfly unit was utilized for Fast Fourier Transform (FFT) and Inverse FFT operations.

The work Chen et al. (2019) which is a representative of the MK-CKKS variant designs an aggregated private key so that only with this private key can a user obtain the plaintext from the final aggregated ciphertext. For a multi-key system with k users, there are k secret keys, $s = s_1, ..., s_k$, the aggregated private key is concatenated by these k secret keys, $\overline{\mathrm{sk}} = (1, s_1, ..., s_k)$. And for k ciphertexts $c = c_1, ..., c_k$ generated by k users, the aggregated key is used to decrypt the aggregated ciphertext $\overline{\mathrm{ct}} = (c_0, c_1, ..., c_k) \in R_q^{k+1}$, $\left\langle \overline{\mathrm{ct}}, \overline{\mathrm{sk}} \right\rangle = c_0 + \sum_{i=1}^{k} c_i \cdot s_i$. Compared to the original MK-CKKS variant, the most recent

MK-CKKS scheme now supports *t*-tolerance and has achieved significant improvements in performance (Du et al. 2023).

The work Lu et al. (2021) is the representative of the torus-CKKS variant. The CKKS based schemes have the advantages of efficient SIMD and efficient approximate operation while the torus based schemes enjoy better computational performance on the torus structure and the lookup table. The work in Lu et al. (2021) combines the advantages of the both type of schemes, and propose the PEGASUS scheme which can switch back and forth between an RLWE ciphertext and a set of LWE ciphertexts by a pair of packing and unpacking methods with the ability to efficiently evaluate both the linear function and nonlinear function.

2.2 Privacy-preserving machine learning

2.2.1 Different privacy protection techniques

PPML is an approach that combines ML with privacy-preserving techniques to ensure the protection of users's private information during data processing and model training/inference, and the parameters of trained models. From a taxonomic perspective based on application scenarios, the techniques employed in Privacy-Preserving Machine Learning (PPML) can be categorized into DP (Dwork 2006), SMPC (Yao 1982), and HE (Rivest et al. 1978). DP falls under the category of Data Analysis and Statistics, SMPC belongs to the domain of Multi-Party Collaboration, and HE is classified under Encryption and Secure Communication. Table 2 shows the comparison among the three different techniques. Additionally these techniques can also be used in combination, e.g., SMPC combined with DP (Mugunthan et al. 2019; Böhler and Kerschbaum 2021) and HE combined with SMPC (Attrapadung et al. 2023; Bian et al. 2023).

Differential Privacy (DP) is a widely recognized and strict privacy protection technique. This concept was first proposed by Microsoft's Dwork (Dwork 2006). DP noise can be obtained through various mechanisms, such as the Laplace mechanism (Dwork et al. 2006) or the Gaussian mechanism (Dwork et al. 2010). Subsequently, many DP techniques have emerged, including Local Differential Privacy (Erlingsson et al. 2014) and Concentrated Differential Privacy (Dwork and Rothblum 2016). The DP technology makes malicious adversaries unable to infer sensitive information about users even if they know the results released by users. DP has been widely applied in various information security domains, such as data release (Cao et al. 2018) and location information protection (Tedeschi et al.

Table 2 Comparison among different techniques

Name	Taxonomic perspective	Advantages	Disadvantages
DP	Data Analysis and Statistics	High privacy and efficiency	Accuracy affected by noise
SMPC	Multi-Party Collaboration	Suitable for complex computing tasks	High computational and communication complexity
HE	Encryption and Secure Communication	End-to-end communication, ciphertext calculation	Low computational efficiency

2023). By applying DP to ML models, training data can be protected from model reversal attacks when model parameters are released. Therefore, there are many works applying DP to ML models (Mangold et al. 2023; Xu et al. 2023; Guan et al. 2023; Shi et al. 2024). The DP technique perturbs the data with noise, with higher privacy and efficiency, but due to the increase of noise affects the performance of the ML model to some extent, which is usually suitable for environments with weaker computing power.

SMPC originates from Yao's Millionaire Problem, and is mainly used to solve the problem of cooperative computation between a group of mutually distrustful participants to maintain privacy. The underlying cryptographic protocols include oblivious transfer protocol (Rabin 2005), garbled circuits protocol (Bellare et al. 2012), Secret Sharing (SS) protocol (Shamir 1979), Goldreich-Micali-Wigderson protocol (Goldreich et al. 2019), and so on. SMPC protocols focus on applications in efficiently parallel and distributed ML (Knott et al. 2021; Li et al. 2021; Gao and Yu 2023; Chen et al. 2024). In some environments, this approach shows the scalability for learning tasks with hundreds of millions of records (Gascón et al. 2016). However, unlike methods that use DP on the model, these protocols, which protect the privacy of the training data to be used in the learning process, suffer from high communication overhead with a large number of communication rounds, low efficiency, and interactive computation. SMPC can be combined with other technologies, such as HE, to protect machine learning schemes. Juvekar et al. proposed GAZELLE, a secure neural network inference scheme based on HE and SMPC, which can switch between HE and SMPC to enhance the efficiency of secure inference (Juvekar et al. 2018). Falcon is another secure inference scheme that combines HE and SMPC, achieving a secure inference accuracy of 99.26% on the MNIST database, which is 0.26% higher than GAZELLE (Li et al. 2020). Building on the GAZELLE technology stack, Mishra et al. introduced Delphi, which includes a planner for automatically generating neural network architecture configurations (Mishra et al. 2020). Compared to GAZELLE, Delphi can increase the latency of secure inference by 22 times. Jha et al., based on Delphi, proposed DeepReDuce, a secure inference scheme that removes ReLU activations (Jha et al. 2021). Since DeepReDuce is also based on Delphi, it utilizes both HE and SMPC. DeepReDuce balances the number of ReLU activation functions used and the inference accuracy.

The connection between cryptography and ML has been studied for a long time, and it is generally recognized that they are in opposition to each other. In a sense, cryptography aims to prevent the access to information, while ML attempts to extract information from data (Graepel et al. 2012). In the field of ML, one way to achieve confidentiality of user data is to utilize traditional cryptographic methods, but the need for encryption and decryption makes it impractical in the real world because of its very high computational complexity. However, with the development of HE, especially the maturity of FHE, a technique that allows arbitrary operations to be performed on encrypted data without the need for decryption, PPML can be achieved with better security level. HE is a true end-to-end encryption system that promises to fundamentally address the privacy issues of data and models, and it allows users to have better control over their data while benefiting from the computational services provided by remote servers. For example, in a centralized ML application, a user uploads training data with a ciphertext form to a server which trains the model without knowing the user's original training data, thus protecting the user's data privacy. Currently HE has been widely used in ML (Choi et al. 2024; Kim and Guyot 2023; Hijazi et al. 2023). As the performance of hardware devices continues to improve, GPU accelerated computation becomes a common practice in the industry, and the problem of high computation and storage overhead of HE is alleviated. The proposal of more efficient FHE schemes and the development of approximate HE schemes supporting floating-point computation make HE based PPML a hot research topic. In addition, HE based PPML schemes can accomplish model training or inference in a non-mutual or non-interactive way compared with SMPC based PPML schemes, simplifying the communication model as well as the communication overhead, which makes them have broader application scenarios.

2.2.2 Security model

PPML schemes based on HE generally use the curious but honest model as their security model. This model is not only applied in HE based PPML schemes, but also has a wide range of applications in other areas of information security, such as searchable encryption (Li et al. 2021) and retrieval updating scheme (Liu et al. 2023). The model usually assumes that the user, the model owner, and the computational party which may be the model provider or a third party outsourcing the computation, are all curious about the data or the model and want to know the detal of the data or the model, but they follow the established rules to perform model training and inference correctly.

2.2.3 Workflow

The workflow of PPML based on approximate HE primarily consists of the following steps:

Step 1. **Data Pre-processing**: Define a function $p(x)$ that takes multidimensional data d as input and outputs a vector z to facilitate encryption into CKKS ciphertext. This process can be represented as the following equation:

$$z = p(d) \tag{6}$$

Different types of data require different pre-processing methods. For example, feature data may need normalization to scale feature values to the range [0, 1]. Text data should be converted into an vector with word embeddings, such as GloVe (Pennington et al. 2014). Image data also needs pre-processing, often through packing methods like multiplexed packing (Lee et al. 2022a), to transform it into a vector format that is easy for CKKS and ML models to handle.

Step 2. **Parameter Initialization**: The data owner has the pre-processed data z. It initializes the parameters for approximate HE, generate public and private keys, as well as the evaluation key. Publicly share the public and evaluation keys, and use the public key to encrypt z to obtain the ciphertext *ct*, which is then sent to the owner of the ML model.

$$ct = Enc(z) \tag{7}$$

Here, *Enc*(x) is the encryption operation.

Step 3. **Model Training or Inference**: The owner of the ML model performs training or inference on the ciphertext. The forward operation can be defined as a function $f(x)$, and the backfowrd operation of gradient update for the new model can be defined as $g(x, y)$.

Step 3.1. **Secure Inference**: If the task is secure inference, the model owner performs the forward operation as Equation (8) and obtain the encrypted inference result o.

$$o = f(ct) \tag{8}$$

Then, the owner of the data z can obtain the inference result ir by decrypting the received o with its private key as Equation (9)

$$ir = Dec(o) \tag{9}$$

Here, $Dec(x)$ is the decryption operation.

Step 3.2. **Privacy-Preserving Training**: If the task involves privacy-preserving training, the backfowrd operation is required to obtain the gradient update u. Therefore, the new model f(x) can be obtained according to the following equation:

$$f(x) = g(f, u) \tag{10}$$

2.2.4 Popular open source libraries

Open source libraries for PPML can be generally categorized into two types: the general-purpose privacy computing frameworks, which implement PPML schemes based on multiple techniques, defined as Type I, and the targeted frameworks, which implement PPML schemes with only one single technique, defined as Type II. The details are shown in Table 3.

There are four main open source libraries in Type I. FATE is an industrial grade FL framework developed by the AI team of Tencent WeBank, which implements privacy computing protocols based on SMPC, HE and so on. Compared with FATE, SecretFlow provides a more complete ecosystem, which is a generalized privacy computing platform provided

Table 3 Popular open source libraries of PPML

Name/Owner	Type	Language	Star
PySyft/OpenMined	I	Python	9.2k
FATE/FederatedAI	I	Python	5.5k
SecretFlow/secretflow	I	Python	2.2k
tf-encrypted/tf-encrypted	I	Python	1.2k
CryptoNets/microsoft	II	C#	269
he-transformer/IntelAI	II	C++	163
CrypTen/facebookresearch	II	Python	1.4k
opacus/pytorch	II	Python	1.4k
Rosetta/LatticeX-Foundation	II	Python	1.4k
jax_privacy/google-deepmind	II	Python	84

by Ant Financial, providing AI and business intelligence workflows and frameworks on its upper layer and implementing security services at the device level on its lower layer, so as to provide implementations of HE, TEE which is short for Trusted Execution Environment, DP, etc. PySyft is a library developed by the OpenMined community for privacy data training which implements privacy computing with FL, DP and HE. And based on TensorFlow using Keras interface for model training and inference on ciphertext, it provides both SMPC and HE based privacy computing.

The other libraries in Table 3 implement only one particular privacy-preserving technique. And for each privacy-preserving technique, there are two open source libraries for DP, SMPC, and HE, respectively. In terms of DP, the opacus library provides functionalities to train DP based PPML models with PyTorch, while the jax_privacy library which is developed by the google-deepmind team also implements DP based PPML schemes. CrypTen and Rosetta are two libraries that utilize SMPC to implement PPML. PPML libraries based on HE are more relevant to this study, where CryptoNets and he-transformer are two libraries developed by Intel for neural network training and inference based on HE. The latter has already implemented CKKS based PPML using the SEAL library.

Moreover, in a narrow sense, FL without DP, SMPC and HE is itself a PPML paradigm, and there are many frameworks available. For example, the *federated* library based on TensorFlow, which is very popular in the open source community, is implemented by the Python programming language, and has a current star number of 2.3k.

3 Technical applications of approximate HE based PPML

In this section, we review the related works from four technical applications, categorized according to a taxonomic perspective based on data types, ranging from simple to complex, including feature processing, text processing, image processing, and audio processing.

Additionally, from another taxonomic point of view, ML can be categorized into supervised learning, unsupervised learning, and reinforcement learning (Muhammad and Yan 2015). Supervised learning can further be divided into traditional models and neural network models. Furthermore, according to a taxonomic perspective based on phases, ML can be divided into two types: training and inference. In this work, we focus on these two perspectives when organizing the related works in the four technical applications.

Finally, based on these four technical applications, we also review the related works from the perspective of three advanced applications.

3.1 Feature processing

In ML, feature processing or recognition is the process of transforming input data into useful information. Features are quantifiable attributes or characteristics extracted from the raw data, which is claasified by algorithms into different classes. An example of feature processing is the classification of the Iris dataset. This dataset is a well-known ML dataset that includes 150 samples, each with four features and a target value representing one of the three varieties of iris. Traditional ML algorithms, such as decision trees, Support Vector Machine (SVM), and logistic regression, can be used for classification tasks on the Iris dataset. The feature processing typically includes data preprocessing, feature selection, feature extraction, feature construction, and feature representation and encoding. After feature processing is complete, the classification model can be trained using the training data, and the model's performance can be evaluated using the test data. By continuously adjusting the feature selection and extraction methods, the classifier's accuracy and generalization ability can be improved.

Considerable literature exists on HE applied to feature processing, including schemes based on approximate HE. Figure 2 provides some examples of related works, which are Kim et al. (2018a), Kim et al. (2018b), Mihara et al. (2020), Liu et al. (2021), Lu et al. (2021), T'Jonck et al. (2022), Li and Huang (2022), Hong et al. (2022), Rovida (2023), and Koseki et al. (2023), respectively.

There is also a special class of models in traditional ML, such as the K Nearest Neighbors (KNN) algorithm, which does not have a strict training or learning process, and generally directly partition the feature vector space of the training data, and use the partition result as the final model of the algorithm. For example, in KNN, the similarity between the target sample and each data in the dataset is calculated to obtain k samples with the highest similarity. And according to the labels of these samples, the one with the highest score of occurrences is generally taken as the label corresponding to the final target sample.

Feature processing is a relatively simple task in ML, and in addition to those done using traditional models, there are currently a number of schemes that use neural networks for feature processing.

There are mainly supervised learning and unsupervised learning algorithms in the technical application of feature processing. The privacy-preserving traditional models of supervised learning contains four main categories: logistic regression, KNN, SVM, and decision trees. While the privacy-preserving neural network models contain two categories: shallow neural networks and extreme learning machine.

3.1.1 Supervised learning

Logistic regression was the first model to be combined with approximate HE. Kim et al. proposed the first privacy-preserving logistic regression model using approximate HE (Kim et al. 2018a), and found that HE is one of the candidates for secure outsourced computation, but its direct application to ML is challenging. This is mainly because the addition and

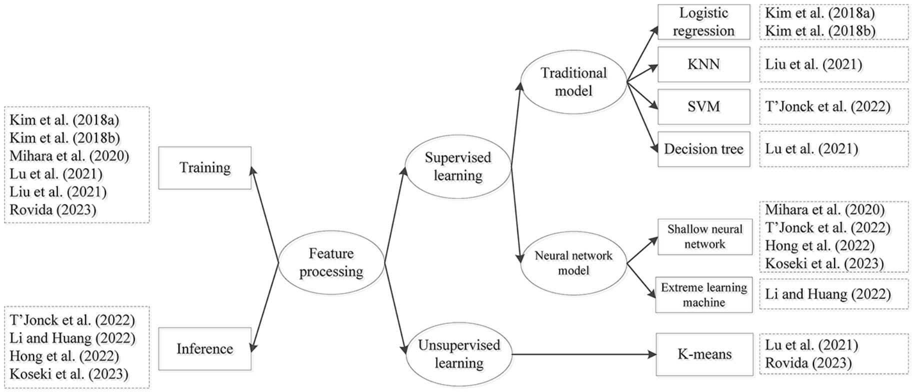

Fig. 2 The classifications of related works on feature processing

multiplication supported by HE cannot directly handle the activation functions needed for ML. For example, the logistic function is required by the logistic regression model. This work Kim et al. (2018a) uses the least squares method to approximate the logistic function, which improves the accuracy and efficiency of the model training. In addition, the logistic regression training generally requires the use of the Newton-Raphson (Ypma 1995) or the gradient descent (Bottou 2010) method, and the Newton–Raphson method involves matrix transposition, which is costly to be implemented by homomophic operations. Therefore, this work uses the gradient descent method, which is then optimized by the packing and parallel techniques. The scheme is implemented on several datasets. For example, training a logistic regression model based on the Edinburgh dataset took 116 min, but yielded fairly accurate predictions. The work in Kim et al. (2018b) also focus on the logistic regression model and optimizes the work Kim et al. (2018a) by designing a new encoding method to efficiently process the entire dataset as one ciphertext in a row-by-row way, compared with f ciphertexts in Kim et al. (2018a), where f is the number of features. In addition, since the ordinary gradient descent method has the problem of zig-zagging on the local optimization path, this scheme uses Nesterov's accelerated gradient method, a momentum variant, to get less iterations and computational overhead while maintaining the same accuracy. The scheme won the title of the best scheme at iDash 2017 track 3. And in terms of efficiency, it can build a privacy-preserving logistic regression model with 1579 items each of which contains 18 features in less than 6 min. The scheme is also trained on five datasets and greatly outperforms the work in Kim et al. (2018a) in terms of the number of iterations, encryption time, storage overhead, and training time. The work in Liu et al. (2021), on the other hand, presents a privacy-preserving KNN scheme. Traditional KNN requires users to send plaintext data to the server for comparison, which is not feasible for some sensitive data with high security level requirements, such as commercial confidential data, medical privacy data, and national security data. This scheme implements a secure KNN classification algorithm with three different similarity calculations: Euclidean distance, Pearson similarity, and cosine similarity, for comparative analysis. For the problem that the CKKS scheme cannot compute the division operation directly, this scheme adopts a client-side computation method, i.e., the ciphertext is sent back to the client, and the client decrypts the ciphertext and then performs the computation on the plaintext. The experimental results show that on the Iris dataset, the accuracy of the other two implementations of the privacy-preserving algorithms is around 97%, except for the Pearson correlation coeffcient implementation which is only around 65%. In addition, the experiment compares the CKKS based scheme with the Pailliar based and the plaintext ones. In terms of computational efficiency, the CKKS based scheme is much faster than the Pailliar based one and is closer to the plaintext one. In terms of storage overhead, the CKKS based scheme, on the other hand, is much better than the Pailliar based and the plaintext ones. In Lu et al. (2021), Lu et al. proposed a privacy-preserving decision trees scheme. In order to efficiently and homomorphically evaluate polynomial and non-polynomial functions, the authors proposed the PEGASUS framework for switching between CKKS-based packed ciphertexts and FHEW-based ciphertexts to process polynomial and non-polynomial functions without decryption. The framework combines "Slots To Coefficients" and "Coefficients Extraction" to pack RLWE ciphertexts into LWE ciphertexts, which synthesizes SIMD, lookup tables, and approximate decryption. The experimental results show that the key size after transformation is very small, decreasing from 50GB to 12MB. To verify the applicability of the framework, the work applies the

PEGASUS framework into the decision trees and the K-means clustering models. Three datasets are used for training the decision trees model, which is then compared with the other two schemes (Lu et al. 2018; Tueno et al. 2020). The experimental results show that the proposed scheme does not have computational advantage, but greatly outperforms the other two schemes in terms of communication overhead achieving an improvement greater than 71x, and also in terms of accuracy. The work in T'Jonck et al. (2022) proposes a privacy-preserving SVM scheme for healthcare scenarios. In order to protect the privacy of medical data, the authors implemented two models SVM and a shallow neural network using the CKKS schemes, and approximated the activation function using linear functions. The experiments are carried out for the Iris and Breast cancer datasets, and the accuracy of the SVM is around 84% and 67% for the two datasets, respectively.

Approximate HE based privacy-preserving neural network schemes are dominated by shallow neural network models and mainly contains four related works Mihara et al. (2020), T'Jonck et al. (2022), Hong et al. (2022), Koseki et al. (2023). Observing that HE has been used in traditional models, but there are fewer related works in deep learning, Mihara et al. addressed the issue how to protect the privacy of shallow neural network models (Mihara et al. 2020). For approximate HE, they on the other hand observed that multiplication and rotation operations on packed ciphertexts are important but computationally overhead, pointing to the design of more efficient packing methods. They analyzed two algorithms for efficient matrix–vector multiplication operations: row packing and diagonal packing. For the computation of transposed weight matrices in backpropagation, which has not yet been solved by HE for neural network training, the authors used the diagonal packing method with pre-rotated "steps" for matrix transposition, which requires only M rotations compared with other schemes. For example, the row packing method requires N multiplications and NM rotations, and the diagonal packing method requires qM rotations, where $N = qM + r$ and r is the residue. This scheme (Mihara et al. 2020) is trained on a three-layer neural network with one hidden layer based on the Iris dataset, and the overall training time is about 29.8 h under 400 iterations. Compared with the plaintext-trained model, there is a slight increase in loss, 0.0249 of plaintext model and 0.273 of ciphertext model, and a slight improvement in accuracy, 0.9805 of plaintext model and 0.9847 of ciphertext model. The work in T'Jonck et al. (2022) implements a shallow neural network with two hidden layers as well as an SVM model, both of which are trained on ciphertexts. The experiments are conducted on the Iris and the breast cancer datasets which show the similar accuracy of the shallow neural network and the SVM model, e.g., both around 96% on the Iris dataset. Medical data is one of the data that needs to be protected in a crucial way, and there is a special competition, iDash, for this purpose. The scheme in Kim et al. (2018b) is one of the outstanding solutions in 2017, and the first track of iDash 2020 is to perform HE-based secure multi-label tumor classification. In Hong et al. (2022), Hong et al. implemented a single-layer neural network using CKKS and used a softmax activation function for multi-classification. Given that HE schemes require polynomialization of the activation function, the authors used the Goldschmidt's divison algorithm (Goldschmidt 1964) for the inverse operation in the softmax function. Experiments are carried out based on The Cancer Genome Atlas dataset and a neural network model is trained on plaintext for subsequent inference on ciphertext. The experimental results show 0.988 of microAUC and 85% of accuracy, wining for co-first place in the first track of iDash 2020. In Koseki et al. (2023), the authors observed that HE for multiplication operations is very expensive, especially when some operations rely on a large number of multiplication operations, such as the approximation of activation function. Therefore, they introduced Stochastic Computing (SC) and proposed the HESC scheme. Two additional operations are added compared with the original CKKS scheme: the Binary to Stochastic (B2S) number transformation of plaintext and encryption/decryption, and the Stochastic to Binary (S2B) number transformation of the decrypted plaintext. HESC utilizes the features of SC to perform homomorphic stochastic operations, where stochastic additions and multiplications can be realized using random multiplexing and bit-parallel logic operations, respectively. The scheme implements a shallow neural network on the Iris dataset, and the implementation results show that the CKKS based HESC scheme reduces the overall time to less than half, i.e., from 98.002 ms to 40.269 ms, although it adds two additional operations: B2S and S2B, and spends more time on decryption, 25.527 ms compared with 0.656ms of the plaintext model. The experimental results also illustrate that CKKS is superior to BFV in this experimental setting, and the CKKS based HESC scheme is superior to the BFV based HESC one.

Neural networks can be classified into two types according to the method of updating model parameters: Extreme Learning Machine (ELM) and deep learning, the former of which does not utilize gradients when updating parameters, still utilizing some weights that are more computationally efficient. In Li and Huang (2022), Li et al. introduced CKKS into ELM and proposed a privacy-preserving ELM scheme, named as CKKS-ELM. The experiment is based on a three-layer neural network with hidden layers, with the Tanhre activation function, which combines the advantages of Tanh and Relu and is approximated by the least square method. The experimental results show that CKKS-ELM is more accurate compared with the other two schemes: HOMO-ELM (Wang et al. 2020) and PP-ELM (Kuri et al. 2017). For example, with the least square method of order 7, the accuracy of CKKS-ELM on the Iris dataset is 97.65%, compared with 89.55% of HOMO-ELM.

3.1.2 Unsupervised learning

The current PPML models implemented with approximate HE for unsupervised learning are mainly K-means. The previously introduced work Lu et al. (2021) implements both decision trees of supervised learning and K-means of unsupervised learning. In addition, the work in Rovida (2023) also implements a privacy-preserving K-means model. The authors observed that with the HE technique, it becomes feasible to perform computations on ciphertext data. Then a secure K-means scheme based on CKKS is proposed and approximate sgn function is utilized to check whether the scheme can be applied in practical scenarios. The scheme also masks the ciphertext data, and the client only needs to execute the lightweight masking algorithm at each iteration, putting computational overheads to the more computationally powerful server side. For example, in the Iris based clustering experiment, the computational overhead of the client side accounts for only 1.48%, while the server side, which is more computationally capable, accounts for 98.52%. The experimental results show the high efficiency of the scheme, with the training/clustering phase completed in seconds, and the prediction/classification operation in the order of one-tenth of a second. At the same time, the accuracy of the scheme is also acceptable, with accuracy greater than 98% for all the eight datasets.

3.1.3 Training/Inference

From the perspective of training and inference, in general, the traditional model basically implements privacy-preserving training. However, for the KNN algorithm, which is based on instance learning and belongs to lazy learning, it does not show the learning process, i.e., there is no training phase. Therefore, the related work Liu et al. (2021), realizes the inference in a privacy preservation way. As for the neural network based schemes, due to the relatively larger computational overhead of this type of model, all of them, except (Li and Huang 2022), only perform the privacy-preserving inference, i.e., secure inference.

3.2 Text processing

Text processing is one of the important tasks in the field of NLP. It contains many tasks, such as text classification, similarity computation, sentiment analysis. Figure 3 shows some representatives of approximate HE based text processing schemes, which are Al Badawi et al. (2020), Podschwadt and Takabi (2020), Podschwadt and Takabi (2021), Lee et al. (2022b), Kim et al. (2022), Walch et al. (2022), Ali et al. (2022), Jang et al. (2022), Wang and Ikeda (2023), and Li et al. (2023), respectively.

Text processing usually requires the use of two types of models, one for word representation models and the other for vector processing models. The former processes words into vector representation, which is convenient for ML models to carry out subsequent learning. In privacy-preserving text processing schemes, this type of models, are generally processed in the client side with the form of plaintext. The latter vector processing model is the one that actually performs the text processing task. Privacy-preserving techniques, such as HE, are generally integrated into this type of models, either for privacy-preserving training or for privacy-preserving inference. For the presentation of the latter models, we also review the related works according to the traditional models and neural network models under supervised learning.

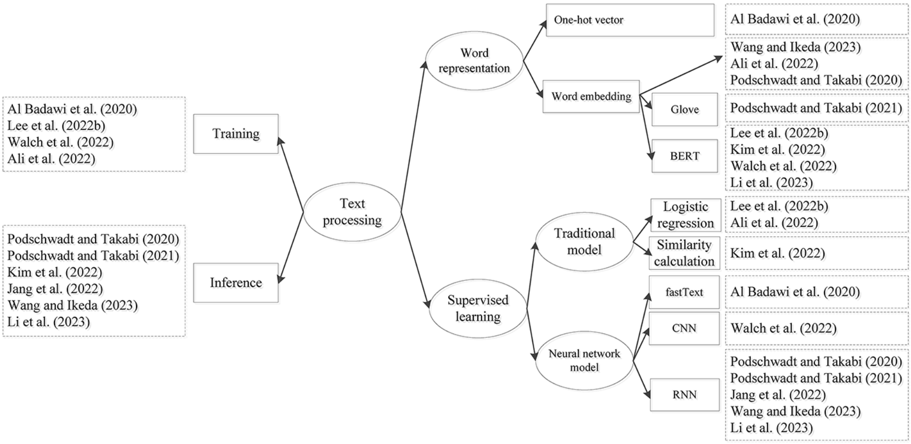

Fig. 3 The classifications of related works on text processing

3.2.1 Word representation

The work in Jang et al. (2022) conducts text processing tasks in the character level, rather than the word and sentence levels. In addition to this work, in other related works, texts are splited into words which are further converted into a vector for subsequent processing. There are two main techniques: one-hot vector and word embedding. For a text with n words, one-hot vector generates a $1 * n$ vector, and each word in a certain index is set to 1, the remaining positions all take the value of 0. Word embedding is a more advanced representation, in addition to being able to convert words into a vecotr with real numbers that can reflects a certain relationship between the words. It avoids the data shuffling attributes of a large number of words, reduces the dimensionality of the data, and adds relationships between words, so as to be able to obtain a representation at the sentence level.

In Fig. 3, only Al Badawi et al. (2020) uses the one-hot vector model, while the other schemes all use the word embedding technique. The work in Podschwadt and Takabi (2020) lists some commonly used pre-trained word embedding models, but does not specify which one is used. The work in Ali et al. (2022) uses the method of Podschwadt and Takabi (2020), but does not specify the used one neither. Although, the work in Wang and Ikeda (2023) does not specify the used word representation model, it can be seen from its Figure 10 that it is based on word embedding model. Other schemes explicitly use word embedding techniques, mainly containing GloVe and BERT models or variants of BERT. For GloVe, there is only one related work Podschwadt and Takabi (2021). The rest of the schemes are BERT model based or BERT variants based, including the basic BERT model (Li et al. 2023), the Sentence-BERT model (Lee et al. 2022b); Walch et al. 2022) and the DistilBERT model (Kim et al. 2022).

3.2.2 Supervised learning

The approximate HE based privacy-preserving text processing mainly contains two traditional models, logistic regression (Lee et al. 2022b; Ali et al. 2022) and similarity computation (Kim et al. 2022). Lee et al. used the Sentence-BERT model to extract sentence embeddings, and then combined the parallel GPU-optimized CKKS scheme and logistic regression model to propose a PPML scheme for binary classification on the Twitter dataset and for multi-class classification on the SNIPS dataset, respectively (Lee et al. 2022b). In terms of the conducted experiments, privacy-preserving logistic regression model is compared with ordinary logistic regression trained on plaintext. The experimental results show that the privacy-preserving logistic regression model on the SNIPS dataset achieves about 98.79% of the F1 metric and 99.58% of the AUC metric, respectively, against 98.76% and 99.89% of the model trained on plaintext, respectively. In addition, this scheme costs only 460.81 s using a single GPU to obtain a test accuracy of 90.8%, compared with PrivFT (Al Badawi et al. 2020) which takes 5.04 days using 8 GPUs to obtain a test accuracy of 86.3%. Therefore, the scheme is not only much more computationally efficient, but also enjoys better accuracy. In Ali et al. (2022), the authors observed that during the Covid-19 pandemic, the Internet was flooded with all kinds of misinformation, and with the increasing privacy and ethical requirements, it became urgent to utilize ML to detect such misinformation. In this regard, the authors proposed an encrypted FL framework, for misinformation detection in a privacy-preserving manner, which consists of privacy-preserving model training and inference, the former for protecting the training data, and the latter for protecting the data privacy of the client user. The authors presented three issues: word embedding and nonpolynomial operations are incompatible with HE, only a certain range of data is supported, and the error keeps increasing as the number of layers increases. To address these three issues, a series of strategies are proposed: using MLP and Self Attention Network (SAN) for misinformation detection; for the SA layer, replacing softmax with sigmoid and further polynomializing it; L2 regularizing the model weights to a certain range, and using sigmoid instead of ReLU to keep the data under control at the range of [0,1]. Experiments are evaluated with three classification models, logistic regression, MLP and SAN, on two publicly available datasets. In order to better control the error generated by the HE scheme, the authors used ReLU instead of sigmoid, reducing the error of the output from 1750 times to 43.75 times. The experimental results on logistic regression show that the accuracy of the trained model on ciphertext is slightly lower than the standard one. For example, the classification accuracy of the encrypted one on the Whatsapp Misinformation dataset is 81% while that of the standard one is 82%. In Kim et al. (2022), the authors paid attention to inversion attacks in text processing, and they found that it is still possible to recover text content from word embedding through inversion attacks, which causes privacy leakage. Therefore, this work not only proposes a privacy-preserving word embedding similarity scheme, but also investigates how to prevent inversion attacks. For inversion attacks, the authors chose black-box inversion attacks, which only accesses the word embedding but not knows the model, and they used 1-layer MLP to extract meaningful information from the word embedding, and their experiments show that the use of d_χ-privacy can protect the privacy to a certain extent but cannot completely prevent the leakage. For example, in the tests on the FIQA-2018 dataset, when $\eta = 150$, the model can recover 5 out of 6 words. Therefore it is necessary to utilize HE to protect the privacy of word embeddings. For the two specific tasks involved in similarity computation, the scheme uses cosine similarity. And for the RNS-CKKS (Cheon et al. 2019) which cannot directly compute the square root inverse function, Newton's method is used Panda (2021) for polynomial approximation. The experimental results show that for the Semantic Textual Similarity (STS) task, the RNS-CKKS based scheme is basically close in effect to the plaintext based scheme and significantly better than the d_χ-privacy based scheme, e.g., for STS-12, 75.296% of Spearman's correlation score (Conneau and Kiela 2018) for the plaintext based scheme and 75.298% for the RNS-CKKS based scheme, whereas only 51.755 25.176% for the d_χ-privacy based scheme when η from 175 to 50. The experiments targeting the text retrieval task have obtained essentially similar results.

Using neural networks to conduct approximate HE based privacy-preserving text processing mainly contains three types of models: fastText (Joulin et al. 2016), CNN and Recurrent Neural Network (RNN), among which RNN is the most widely used neural network model in this technical application. The fastText model is a text classification model which uses a shallow neural network to achieve word2vec and text classification, and its effect is similar to the deep network, saving resources and having a hundredfold speedup. Therefore, it is also the only neural network model that implements privacy-preserving training in text processing. In Al Badawi et al. (2020), the authors observed that although MLaaS becomes very useful, there are two main challenges in this paradigm: one is evaluating user inputs with a cloud-server model, and the other is training a model through a cloud server. To address these two scenarios, the authors proposed PrivFT, short for Private and Fast Text classification, which provides the ability to secure inference on ciphertext with a trained model on plaintext, and also implements the training of model on encrypted data. The authors implemented a 1-2 orders of magnitude optimization using the GPU-optimized CKKS scheme. Compared with the SEAL library, the performance improvement for each operation is 11.91 to 224.62 times. The experimental results show that the GPU-optimized CKKS scheme based on the fastText model takes 0.17 s to make secure inference on encrypted data and 5.04 days to train a model using encrypted data.

The initial development of CNN lies in the technical application of image processing. With the deepening of its research, it also shows very good performance in text processing. In Walch et al. (2022), the authors observed that for some application scenarios lacking data support, Transfer Learning (TL) is generally used for these scenarios, but TL (Fuzhen et al. 2021) also suffers from privacy leakage in some application scenarios, such as model inversion attacks (He et al. 2019). For this reason, the authors proposed CryptoTL mainly containing two models, the server side model and the client side model. The server side has its own data, defined as D_s, used to train a server-side model, defined as M_s, mainly using DP, specifically the differentially private stochastic gradient descent (Abadi et al. 2016) optimization. Then the client side has a small amount of data, defined as D_c, which is encrypted using HE and is sent to the server side to make secure inference, and the returned inference results are used to construct the clent model defined as M_c. In addition, cryptoTL can be used to classify data for the client side only. The authors implemented a CNN based cryptoTL model with the ReLU approximation using the Matlab's toolbox Chebfun (Driscoll et al. 2014). This model is ultimately used for two text processing tasks: sentiment analysis and spam detection. Due to its use of CNN, this model can also be used in other application scenarios, such as image recognition. The experimental results show that compared with PrivFT, CryptoTL has lower accuracy but is better than the baseline CNN model on IMDb dataset, which are 89.88%, 86.29% and 85.83%, respectively. Compared with the baseline CNN model on the other two datasets, CryptoTL is better on YouTube, the accuracies of which are 96.67% and 93.96%, respectively. While the baseline CNN model works better on the Twitter dataset, the accuracies of the two models are 87.28% and 86.70%, respectively.

The RNN model is a kind of neural networks with a recurrent structure capable of processing sequential data and retaining the memory of previous information. Variants of RNN include the Long Short-Term Memory Network (LSTM) model and the Gated Recurrent Unit (GRU) model. The LSTM model introduces three gating mechanism: input gate, forgetting gate, and output gates, and a cellular state dealing with the longterm dependencies more efficiently. While the GRU model, on the other hand, combines the input and forgetting gates into an update gate, which simplifies the structure but performs well on some tasks and reduces the number of parameters, making training faster. These variants improve the efficiency and performance of RNN in processing sequential data. Therefore, RNN and its variants have been widely used in NLP, such as machine translation (Zhao et al. 2022), and also in some multimodal tasks related to natural language, such as generating image descriptions (Karpathy and Fei-Fei 2017). In Podschwadt and Takabi (2020), Podschwadt and Takabi proposed a PPML scheme that combines RNN and HE for secure inference of NLP tasks on encrypted data. A method is proposed to optimize the encryption of word embeddings by directly using pretrained embeddings, such as GloVe (Pennington et al. 2014) and Bert (Devlin et al. 2018), and then to share these embeddings to the client. In this scheme, the server uses a pre-trained model to conduct privacy-preserving inferences, and the client sends its encrypted data and gets the encrypted inference result from the server. The experimental results show that the accuracy of the inference and the accuracy of the model are consistent, both are 86.47%. The scheme costs additional communication overhead due to the form of client-side computation when dealing with the activation function. The work in Podschwadt and Takabi (2021) is aimed at carrying out text processing task without the server-client interaction. This scheme uses the word embedding GloVe and a parallel RNN structure which is more suitable for HE, reducing the depth of RNN by decreasing the length of the input sequence which leads to information loss. Based on the observation that shorter sequences reduce the multiplicative depth of RNN, the authors proposed an alternative solution to run the RNN on encrypted data by splitting the input sequence into shorter subsequences of equal length which are then fed into the RNN layers, concatenating the outputs of the RNN layers, and then feding into the fully connected layer. The experimental results show that the F1 score of this PPML model on two datasets, the online product reviews dataset and the IMDb movie reviews dataset, are 88.8% and 74.36%, respectively. It is within 3% when compared with the benchmark, thus the scheme is 4.

In order to perform matrix representation and manipulation more efficiently and to control the growth of noise, a CKKS variant based on the multivariate ring learning with errors (m-RLWE) problem, called MatHEAAN, is proposed, combining with the GRU model (Jang et al. 2022). The training of the model in this scheme is performed on the plaintext, but consists of two rounds, in order to obtain a pre-trained model compatible with the approximate activation function. In the first training round, the original activation function is used and the intermediate pre-trained values are stored for polynomial approximation. L2 regularization is used in this round to control the weight range. Afterwards, a second training round is performed and the pre-trained approximation polynomials are used to re-train a model whose weight parameters are encrypted and subsequently used for the weights of the MatHEFC (Matrix Homomorphic Encryption Fully Connected) layer and MatHEGRU (Matrix Homomorphic Encryption GRU) layer for the later secure inference. The experiments of this scheme are carried out for two application scenarios: image processing, and text processing which includes Sequence Copy and Genomic Sequence Classification. For the Sequence Copy task, both the plaintext and ciphertext models obtain 100% accuracy, i.e., there is no loss of accuracy in the ciphertext model. For the Genomic Sequence Classification task, both the plaintext and ciphertext models obtain 89.7% accuracy, also meaning that no loss of accuracy in the ciphertext model.

In Wang and Ikeda (2023), the authors observed that there have been many current researches applying HE to forward neural networks, but there are fewer researches on RNNs incorporating HE because deep recurrent operations are difficult to combine with HE. Therefore, they used some techniques to reduce the multiplication depth in each recurrent step to 8, while only 7 recurrent steps are needed in a model. Due to the reduced multiplication depth, this model does not require bootstrapping when it is applied to inference. In addition, the authors found that the temporal segmentation and rearrangement of the input sequence does not reduce the accuracy, i.e., the GRU model is still able to extract the overall time-related features from each segmentation portion. Therefore, they designed three different types of GRU neural networks by temporal segmentation and rearrangement, two of which were applied to text processing. Meanwhile, Layer Normalization (LN) is used in order to reduce the range of the input data. The experimental results show that this model obtains an accuracy of 90.0% on the AG-news dataset, which is slightly lower than 90.3% of the plaintext GRU model.

In Li et al. (2023), the authors found that fusing multimodal data into a single model requires the use of tensor fusion network, but the high-dimensional Cartesian product it requires cannot be directly combined with HE. To address this issue, the authors used a Packing One Modality optimization that encodes a single modaility into a ciphertext and multiplies it with the elements of other features. To further optimize the efficiency, an extension technique, pre-expansion, is proposed to extend the various representation to the length of the fusion before encryption, which makes it possible to fuse two modals with only one multiplication. In addition, they proposed a packing technique to pack *n* ciphertexts into a single ciphertext through consecutive rotation operations. The experimental results show that the secure inference by the model does not cause accuracy loss on both datasets, CMU-MOSI and SIMS, and the F1 scores of the model on the two datasets are 74.67% and 76.97%, respectively. The scheme that combines pre-expansion and packing technique achieves a data throughput of 211.88 KB and the computation time is 3.14s, compared with the use of CKKS alone which are 6.03MB and 149.21s, respectively. In addition, the further use of GPU acceleration makes the computation time drops to 0.91s when the data throughput remains unchanged. Comparison experiments with CryptoNets show that the scheme has a good performance in multiplication depth, input size, output size, and encryption/decryption time, obtaining a speedup from 28x to 272x.

3.2.3 Training/Inference

As shown in Fig. 3, there is a greater focus on secure inference rather than training in the technical applications of privacy-preserving text processing. Among the four relevant works that concentrate on training privacy-preserving models, the primary model pertains to logistic regression in traditional models (Lee et al. 2022b); Ali et al. 2022). The remaining works are on fastText (Al Badawi et al. 2020) and CNN (Walch et al. 2022). Although, the work in Walch et al. (2022) also implements privacy-preserving training, employing a method distinct from others based on approximate HE schemes; instead, it utilizes differential privacy (DP). Additionally, the work in Al Badawi et al. (2020) involves training a model from scratch with an encrypted dataset.

In terms of inference, except for Kim et al. (2022), which focuses on similarity computation, the rest are neural network models. Most of them are RNNs and their variants, demonstrating that while RNNs have been widely used in the field of text processing, introducing privacy preservation based on approximate HE can currently only be achieved for inference, not for training models with higher computational requirements.

3.3 Image processing

Image processing plays a crucial role in ML. The shift from traditional ML models to deep learning has driven significant advancements in image processing. This field finds wide application across various domains, particularly in face recognition. The development of image datasets is intricately linked to the evolution of image processing algorithms. From the classic MNIST dataset to CIFAR and the extensive ImageNet dataset, these datasets play pivotal roles in model development for image processing. MNIST, featuring grayscale images of handwritten digits, remains a cornerstone dataset in this domain. SVMs and Random Forests are commonly employed for image classification tasks on MNIST, utilizing manually crafted features and traditional classifiers for training and predictions. The CIFAR dataset comprises 10 different categories of color images and poses a greater challenge compared with MNIST. With the rise of deep learning models, CNNs have emerged as powerful tools for automatically learning feature representations from data. Models like LeNet, AlexNet, and VGG have become dominant on the CIFAR dataset. These CNNs extract features from images using multi-layer convolution and pooling operations and classify them through fully connected layers. The ImageNet dataset, which includes millions of high-resolution color images across 1000 categories, marks a significant milestone in image datasets. Deep learning models achieve a breakthrough in the ILSVRC competition for ImageNet. This phase witness the introduction of more complex and deeper models such as GoogLeNet, ResNet, and Inception. These models employ deeper network architectures and more intricate feature extraction methods to achieve superior performance in image classification, leveraging large scale training data and computational resources.

Moreover, current image processing models based on approximate HE have gained significant momentum. While image processing is inherently more complex than text processing, the application of approximate HE to image processing predates its application to text processing. Figure 4 presents some representative works on image processing utilizing approximate HE schemes, which are Jiang et al. (2018), Boemer et al. (2019), Ishiyama et al. (2020), Jung et al. (2021), Lee et al. (2022c), Lloret-Talavera et al. (2022), Li et al. (2022), Sperling et al. (2022), Lee et al. (2023), and Y et al. (2023), respectively.

In the technical application of image processing, the current literature only addresses supervised learning. Traditional models primarily include logistic regression, whereas neural network models offer richer alternatives such as CNNs, ResNet, CryptoNets, MobileNetV2 (Sandler et al. 2018), and VGGFace. Furthermore, in such technical applications, there are framework-level algorithms that encompass the graph compiler nGraph and information processing frameworks like the Hyper Dimensional Computing (HDC) model.

3.3.1 Framework level

The nGraph framework is a graph compiler for DNNs, developed by Intel, and currently supports frameworks such as TensorFlow directly and PyTorch indirectly. The nGraph-HE framework is a combination product of Intel's nGraph and HE. In Boemer et al. (2019), the authors found that current privacy-preserving deep learning models such as nGraph-HE,

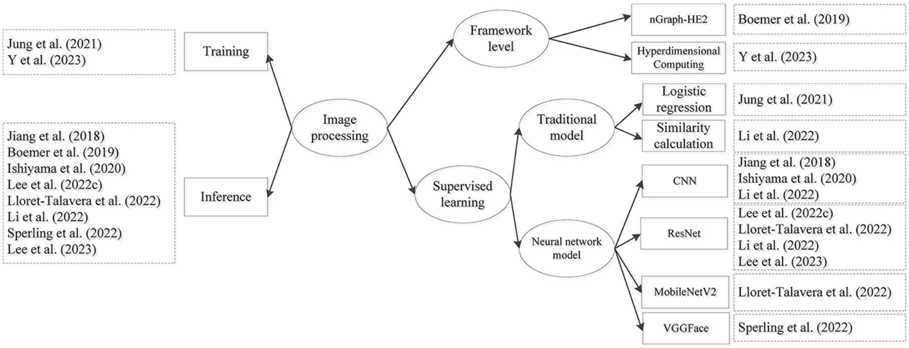

Fig. 4 The classifications of related works on image processing

only support relatively shallow models and polynomial activation functions. The authors proposed an nGraph-HE2 model based on a CKKS version with three main optimizations: coding optimization for CKKS including scalar encoding and complex packing, ciphertext-plaintext addition and multiplication for CKKS, and a graph level optimization including the lazy-rescaling and depth-aware encoding. In addition, for the nonlinear functions in the model, this scheme uses a client-side computation mechanism, i.e., the server sends the ciphertext to the client, the client decrypts the ciphertext, conducts operation on plaintext, encrypts the result, and sends the result ciphertext back to the server, and the server conducts further computation. Experimental results show that the scheme utilizing lazy-rescaling is 8 times faster than the ordinary CryptoNets (Gilad-Bachrach et al. 2016). And the experiments on MobileNetV2 show that the scheme is only about 0.01% less accurate than the original MobileNetV2 model. Moreover, this scheme is used for secure inference on MobileNetV2 and ResNet (Lloret-Talavera et al. 2022).

HDC is an emerging information processing and computing framework based on combining primitive computing unit operators to realize high-dimensional data processing. It offers advantages such as processing complex data, being unaffected by noise, supporting incremental learning, and enabling fast searching. HDC has found wide applications in fields such as text processing, image processing, and pattern recognition. The work in Y et al. (2023) integrates HDC with HE for image processing. The authors observed that existing FHE schemes typically utilize only a single SoC and often neglect scalability. To tackle this limitation, the authors optimized both algorithmic and hardware aspects. At the algorithmic level, HDC is applied to FHE, maintaining higher accuracy with reduced complexity compared with neural networks based on backpropagation. HDC-based models are simpler, more HE-friendly, and tolerant to errors. On the hardware level, the authors devised a specialized ASIC based system architecture with an interconnection network linking multiple HE accelerators. This architecture also implements automatic scheduling and data allocation, enhancing both throughput and memory efficiency. Experimental results demonstrate that this HE-friendly HDC model not only achieves superior accuracy but is also 4.7 times faster in training on encrypted data compared with previous schemes based on Multi-Layer Perceptron (MLP). Furthermore, it outperforms FHE based RNN models by 1000 times in terms of inference speed. Moreover, the scheme excels in data parallel processing efficiency and energy overhead, achieving speedups of 38.2 times and 13.8 times, respectively.

3.3.2 Supervised learning

Logistic regression is the only traditional model used in the technical application of image processing. In Jung et al. (2021), through their analysis, the authors found that the important issue affecting the performance of the CKKS scheme comes from the high main memory bandblock requirements which leads to the use of memory-centric optimization technique. The kernel fusion technique provided by the GPU is utilized to optimize the intra and inter operations, which greatly reduces the amount of memory accesses required by HE, and this GPU-based optimized CKKS scheme achieves a 7.02x speedup. This GPU-optimized CKKS scheme is then applied into the training of a logistic regression model. Experimental results show that this implementation achieves a 40x speedup compared with the previous 8-threaded CPU implementation.

The explosive development of deep learning originated from breakthroughs in the field of image recognition, leading to the creation of numerous excellent neural network models. Similarly, in approximate HE based PPML, there exists a rich variety of solutions for these models, with CNNs standing out as one of the most prominent examples.

In Jiang et al. (2018), the authors observed that HE supports a non-interactive mechanism and has low communication overhead compared with SMPC, but the computational optimization is not well done, especially on matrix computation. Therefore, they designed a matrix encoding method and an efficient matrix homomorphic evaluation strategy, explaining how to encode multiple matrices into a single ciphertext to improve the efficiency, mainly better amortized performance. This strategy can be applied to many HE schemes, and it can achieve 64th-order matrix multiplication and transposition operations to the level of seconds, which are 9.21 s and 2.56 s, respectively. Based on this method, they proposed the E2DM framework, short for Encrypted Data and Encrypted Model, which realizes the inference on ciphertext dataset and model. The experiments are conducted on the MNIST dataset and the secure inference of CNN model is implemented and the results show that for 64 images, it takes 28.59 s and the amortized time for each image is 0.45 s. In Ishiyama et al. (2020), the authors found that the use of low order approximation polynomials, such as the squared function, leads to lower model accuracy. Therefore, they focused their work on a comparative accuracy analysis of activation function. They comparatively analyzed the effect of model accuracy by using polynomials of different orders for two different activation functions, Swish and ReLU. In addition, their proposed scheme uses a batch regularization technique to regularize the input data to minimize the error. The experiments are performed based on the CKKS implementation of the SEAL library and the inference of the privacy-preserving CNN model is implemented. The experiments for the MNIST and CIFAR-10 datasets obtain 99.29% and 81.06%, respectively, which achieve accuracy improvements of 0.11% and 4.69%, respectively, compared with the previous scheme.

ResNet is a variant of CNN, which is proposed by He et al., to address the *degradation* issue in deep neural network training (He et al. 2016). It introduces residual blocks to simplify the training process of the network, and enables deeper networks to be trained efficiently through the techniques such as identity mapping. Related works Lee et al. (2022c), Lloret-Talavera et al. (2022), Lee et al. (2023) are the proposals on privacy-preserving ResNet models using approximate HE. In Lee et al. (2022c), the authors observed that the previous PPML schemes are limited by simple and non-standard models that are not effectively implemented on large datasets, do not use simple functions instead of nonlinear activation functions rather than their approximation polynomials, and do not use bootstrapping. This work uses RNS-CKKS with bootstrapping, implements the standard ResNet-20 model, and implements model training on the CIFAR-10 dataset. In addtion, the work devises an approximation method with very good accuracy, which uses the least squares method to approximate exponentiation operation and the Goldschmidt's divison algorithm to approximate the division operation in the Softmax activation function. If the Softmax input value is too large, it is constrained using Gumbel Softmax function (Jang et al. 2016). The experimental results show that the accuracy of this ResNet-20 model based on approximate HE is 92.43%*pm*2.65%, which is very close to the accuracy of the orignal model with plaintext, and the inference time is within 3 h. In Lloret-Talavera et al. (2022), the authors observed that the inference on encrypted data based on HE consumes 100x-1000x memory and computation overhead. Even small models in PPML require GB-level memory overhead. To overcome this iisue, the authors utilized a hybrid memory model that integrates DRAM and perisitent memory technology, i.e., Intel $\mathrm{Optane}^{\mathrm{TM}}$ Pmen, and also utilized Intel's nGraph-HE2 (Boemer et al. 2019) to run large neural network inferences, such as MobileNetV2 and ResNet. For the ResNet50 model with 2048 batches, the scheme implements a privacy-preserving inference that takes 63 h and requires a memory overhead of 900 GB at peak. In Lee et al. (2023), in order to mitigate the overhead of the client and server when using HE schemes, e.g., CKKS and BFV, a hierarchical rotation method on the secret key is proposed. By sending a small set of rotatable secret keys from the client, the server can generate the public key, which reduces the communication and computation overhead between the two parties and can be computed offline through a secret key transformation process. Through different layers of the rotation key system, the size of the rotation key is reduced by different magnitudes. For example, to realize a ResNet-18 model on ImageNet, the server needs 617 rotation secret keys, and the client needs to spend 145.1s to produce a whole set of 115.7GB rotation keys, whereas using a 2-layer (resp. 3-layer) structure, the whole set of rotation keys can be reduced to 2.91GB (resp. 1.54GB) which takes takes 3.74s (resp. 1. 93).

There are also current shemes that use both traditional models and neural network models, but the two types of models do not serve the same purpose. For example, the work Li et al. (2022) utilizes neural network model to extract features and then uses similarity computation for feature recognition. In Li et al. (2022), the authors observed that with the development of AI, biometric authentication has reached a better level of protection than methods using password, especially in the post epidemic era. However, most biometric authentication algorithms are based on single-feature, which can only provide a certain level of accuracy and are prone to spoofing attacks (Nguyen et al. 2014). Therefore, fused multi-feature biometric authentication has become a hot research topic. For biometric feature processing, the proposed scheme uses Face-CNN and Voice-CNN to extract image and audio features respectively, and then uses ResNet-34 to perform the training on the extracted features and as the aggregation model. In addition, MK-CKKS, a multi-key CKKS variant, is used to build this efficient privacy-preserving multi-biometric recognition scheme, which increases the applicability of the scheme by reducing dependence on the central node through the use of MK-CKKS. Cosine similarity is used for feature recognition. The experiments involve feature extraction and fusion using the WebFace, VoxCeleb2, and MD-XJTU datasets, achieving an Equal Error Rate (EER) metric as low as 0.66%. In Sperling et al. (2022), the authors investigated multi-feature biometric recognition for the same purpose as that in Li et al. (2022), and also proposed a practically feasible multi-biometric fusion and matching algorithm with three components, coding scheme, matrix multiplication, and approximation regularization with a combinatorial polynomial used to approximate the square root inverse function. The multi-feature biometric recognition scheme uses VGGFace to extract face features and Deep Speaker model to extract voice features, and then uses splicing to fuse the two features. In the authentication phase, the scheme also uses cosine similarity as a benchmark. The experimental results demonstrate the high efficiency of the scheme in fusing and matching multi-features in 884ms with 1024 objects. Compared with two separate biometric authentications, the fused scheme improves the AUROC metric by 11.07% and 9.58%, respectively.

3.3.3 Training/Inference

In general, image processing demands high device performance, and it is through the development of high-performance computers that image processing using deep learning has experienced explosive growth. Therefore, combining approximate HE, from the perspectives of training and inference, only two related works have achieved training on encrypted data: one based on the traditional model logistic regression (Jung et al. 2021) and another based on hyperdimensional computation (Y et al. 2023). All privacy-preserving models based on deep neural networks in Fig 4 have thus far only achieved inference on encrypted data.

3.3.4 Other important related works

Image processing is currently one of the hottest areas of ML, and PPML has produced a large amount of research in this technical application. Other important related works include (Lee et al. 2022a; Kim and Guyot 2023), which are not described in detail. These two works mainly perform optimization of homomorphic convolution operation, i.e., optimizing the homomorphic computation in the convolutional layers. The former implements a secure inference of ResNet-100 model, and the latter performs constant convolutional evaluation regardless of the kernel size. Both of the works are implemented on the standard dataset CIFAR10/100 for secure inference.

3.4 Voice processing

Voice processing, also known as voice/speaker data processing, is a crucial application of ML. It involves the automatic identification and classification of voice, speaker data, or music by analyzing audio signals and using models to learn the feature information in this kind of data. Voice processing has found widespread use in both professional and personal settings. Voice assistants like Siri and Alexa use audio recognition technology to understand and execute user commands (Ahmed et al. 2023), such as playing music, sending text messages, and providing weather forecasts. Additionally, voice processing can be utilized for real-time subtitle generation, significantly benefiting individuals with hearing impairments by improving their understanding of TV programs, conference speeches, and other similar content. Moreover, voice processing is commonly employed in music recognition and recommendation systems, allowing users to easily discover their preferred music (Deldjoo et al. 2024).

Traditional voice processing models are typically based on manually designed feature extraction methods, such as Mel-Frequency Cepstral Coefficients (MFCC) and Spectrogram. These methods often require prior knowledge and are difficult to handle complex audio signals. In contrast, deep learning models can automatically extract complex audio features, resulting in more accurate classification and recognition. Currently, commonly used deep learning models include CNN, RNN and Transformer (Vaswani et al. 2017; Devlin et al. 2018). These models are widely used in various fields due to their effectiveness in processing complex data.

Like text or sentence in text processing has to be expressed into vectors, voice data in voice processing has to be expressed into a format that can be easily handled by ML, typically from audio files in wav format into voice feature vectors. Here we refer to this as voice feature expression. At this point, some applications will directly use these feature representation as input to the model. However, there are also some applications that perform further feature extraction on these feature representations. The former is similar to one-hot in text feature representation, while the latter is similar to the word embedding technique. There are traditional feature processing techniques, such as i-vector, as well as neural network based models, such as CNN, which are usually not combined with HE in privacy-preserving voice processing. Here we refer to this feature processing as voice feature extraction. In addition, the vectors generated by feature expression or feature extraction are used as inputs to ML models and combined with HE to become PPML models. This is the part of this work focusing on.

Voice processing is directly facing the user. Therefore, some real world applications of this technical application require more privacy protection for the user. Some typical voice processing literature using approximate HE schemes shown in Fig. 5 are Chindris et al. (2020), Rahulamathavan (2022), Li et al. (2022), Sperling et al. (2022), Zheng et al. (2022), Elworth and Kim (2022), Li et al. (2023), and Zhang et al. (2024), respectively.

We first present the related works in terms of voice feature representation and feature extraction, then provides the detail of the related works according to the based traditional models and neural network models of supervised learning, and finally sort out them from the perspective of training and inference.

3.4.1 Voice feature representation

To perform audio processing with ML models, audio files in special formats such as.wav are first converted into audio feature vectors-representations of audio files as voice features that can be easily processed by ML models. In the field of privacy-preserving audio processing based on approximate HE, three main feature representations are used: Spectrogram (Chindris et al. 2020; Elworth and Kim 2022; Zhang et al. 2024), Filter bank (Sperling et al. 2022), and MFCC (Rahulamathavan 2022). The remaining related work in Fig 5 does not specify how feature representations are conducted.

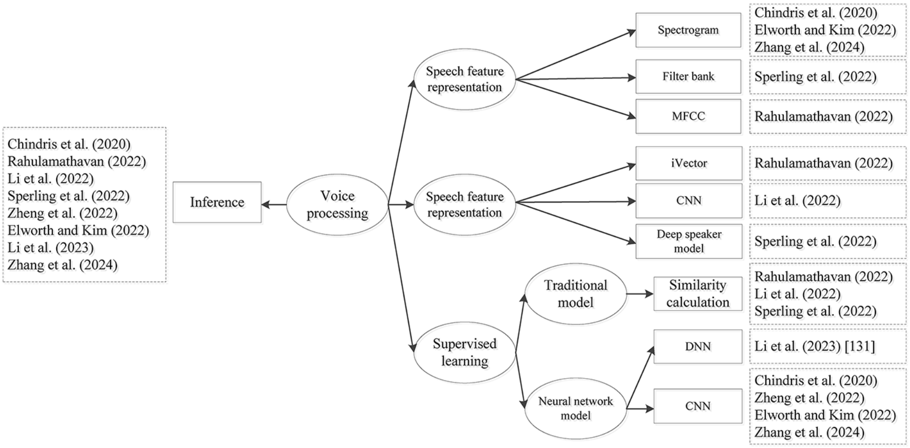

Fig. 5 The classifications of related works on voice processing

3.4.2 Voice feature extraction

After audio files are processed into feature representations, these representations can be used directly in certain application scenarios or further processed. This further processing is generally aimed at extracting audio features from the feature representations. According to Fig 5, it can be observed that in the related works of privacy-preserving audio processing based on approximate HE, there are three studies specifying the feature processing models used: the traditional model iVector (Rahulamathavan 2022), and neural network-based models such as CNN (Li et al. 2022) and Deep Speaker model Sperling et al. (2022).

3.4.3 Supervised learning

In the technical application of voice processing based on approximate HE, supervised learning primarily involves similarity computation in traditional models and two types of neural networks: Dense Neural Network (DNN), where each layer consists of a fully connected layer and activation function, and CNN.

Voice processing is often integrated with and complemented by other types of processing. For example, in Li et al. (2022), the authors employed Face-CNN and Voice-CNN to process image and audio data respectively. They then trained extraction and aggregation models using ResNet-34, concluding with privacy-preserving cosine similarity for recognition. Similarly, in Sperling et al. (2022), the authors used VGGFace and Deep Speaker models to process image and audio data respectively, and then used privacy-preserving cosine similarity to do authentication. In Li et al. (2023), the authors used LSTM to extract text features, and DNN with three hidden layers to extract audio and video. these features are fused and then homomorphic DNN is used for the final inference.

Related works on audio processing alone includes (Rahulamathavan 2022; Chindris et al. 2020; Zheng et al. 2022; Elworth and Kim 2022; Zhang et al. 2024). Among them, the work in Rahulamathavan (2022) conducts the similarity computation and the other four related works focus on CNN. In Rahulamathavan (2022), the authors observed that as more smart devices incorporate built-in automatic speech recognition, there is a significant increase in concerns over user biometrics and personal data privacy leakage, which can be addressed through cryptography. Consequently, the authors redesigned a voice authentication system by integrating HE. Since the similarity calculation for authentication involves cosine distance, the square root inverse function is necessary. However, the CKKS scheme cannot directly handle the square root inverse operation on ciphertext. Therefore, the authors proposed a novel approach using Newton's method, which reduces the multiplication depth with minimal impact on accuracy. Experimental results demonstrate that the authentication system can complete authentication within 1.3 s, enabling real-time authentication with only a 2.8% increase in the EER metric compared with the plaintext model.

There are four related works combining HE and CNN (Chindris et al. 2020; Zheng et al. 2022; Elworth and Kim 2022; Zhang et al. 2024). In Chindris et al. (2020), the authors utilized nGraph-HE to construct an HE based neural network. The system contains three components, HE-Speaker Identifier Client, pyHE Client, and HE-Speaker Identifier Server. The HE-Speaker Identifier Client extracts the spectrogram information and processes it in character form, removes the quiet part, and creates spectrograms for each window in the signal. Thus, a Mel-scale spectrogram is generated. The pyHE Client conducts the encryption and decryption operations, and communicates with the server by the CKKS implementation provided by SEAL. The HE-Speaker Identifier Server optimizes a CNN and incorporates several enhancements, such as optimizing a special communication protocol, refining polynomial approximations for ReLU and Sigmoid functions, and substituting max pooling with average pooling. These improvements elevate accuracy to 99.6% and reduce evaluation time to 340 s. In Zheng et al. (2022), the authors observed that the proliferation of multimedia data has made storing and processing audio data on personal devices burdensome. While leveraging cloud computing and services can alleviate this burden, they also pose challenges to data privacy. Therefore, employing cryptography becomes essential for safeguarding audio data. In previous work, it is generally assumed that features are extracted in plaintext and ciphertext is used only during inference. In this work, the authors further enhanced the privacy protection during feature extraction. A complex-valued CNN model is constructed, including complex convolutional, pooling, and fully connected layers, which can learn more voice representation than a real-valued CNN model, and thus obtains higher accuracy on the Keyword Spotting task. Furthermore, for the complex activation function, the authors proposed two forms and polynomialized them with two approximation methods: least square method and derivative approximation method. The authors constructed a complex-valued CNN model comprising six convolutional layers and employing four approximation activation function methods. On the Keyword Spotting task, this model achieved an accuracy ranging from 73.9% to 74.4%, significantly outperforming other schemes. For instance, CryptoNets achieves an accuracy of only 35.9%, while CryptoDL achieves 17.2%. In Elworth and Kim (2022), the authors proposed a scheme named as HEKWS, short for Homomophic Encryption KeyWord Spotting, which uses a small-footprint CNN model based on the Hont scheme (Tang and Lin 2017) to minimize the parameter or the multiplication depth, and relies on an optimized packing method for the client, where the input of the spectrogram has a shape of (40, 32). Then the input is packed in the frequency dimension so that 1,280 ciphertexts can be packed into 32 ciphertexts. Three optimization techniques are designed for the server side, including masking, inter-slot addition and expansion, to transform the ciphertexts from the client side into a format more suitable for the CNN model. The experimental results show that HEKWS obtains an accuracy of 72.3%, while the Hont scheme achieves 76.0% and 72.1% accuracies, without and with the use of the approximate activation function, respectively. Therefore, it can be seen that HEKWS with HE improves the accuracy by 0.2% over the Hont scheme using the approximate activation function. In addtion, the scheme takes only 19 s to perform a secure inference and subsequent techniques such as parallelism and hardware acceleration are expected to further reduce the inference time. In Zhang et al. (2024), the authors proposed three optimizations to enhance secure voice retrieval services: Optimization 1 utilizes Tri-CNN to extract features from the spectrogram graph and generate hash information for retrieval. Optimization 2 employs segmentation and batch processing to reduce the complexity of encrypting voice data. It incorporates mode conversion and relinearization techniques to mitigate the expansion of encrypted voice data when generating the voice dataset. Optimization 3 introduces an optimized voice retrieval scheme for securely computing similarities. This scheme matches the similarity of retrieved information to facilitate the retrieval of corresponding encrypted voice data. Experimental results demonstrate that the accuracy of voice retrieval exceeds 93% on both datasets, with only a 2% loss compared with the plaintext model.

3.4.4 Training/Inference

Traditional PPML models in technical applications of voice processing focus on similarity computation (Rahulamathavan 2022; Li et al. 2022; Sperling et al. 2022), which do not require a training phase to perform privacy-preserving classification or inference directly. Neural network-based models, on the other hand, utilize plaintext trained models for privacy-preserving inference. Thus, the related works depicted in Fig 5 revolve around privacy-preserving inference. Further development is needed for training privacy-preserving models specifically tailored for voice data.

3.5 Advanced technical applications

In addition to the four previously mentioned technical applications, several advanced applications have emerged: video processing, which integrates image and audio processing; massive text processing, commonly referred to as LLMs; and distributed learning frameworks, represented by FL. In this subsection, we review related works from these three perspectives in accordance with Fig. 6.

From an ML perspective, there exists a strong connection between video processing and image/audio processing. Video can be seen as a sequence of images in the time dimension. Therefore, video processing involves applying image processing techniques and also considering the characteristics of time-series data. In video processing, methods from image processing and computer vision can be utilized to analyze and process each frame, while incorporating audio processing techniques to handle the audio information within the video. ML algorithms enable tasks such as automatic identification, target tracking, and activity detection in video content, combining audio and image processing techniques to achieve a more comprehensive and integrated analysis and understanding of video content. Some cur-

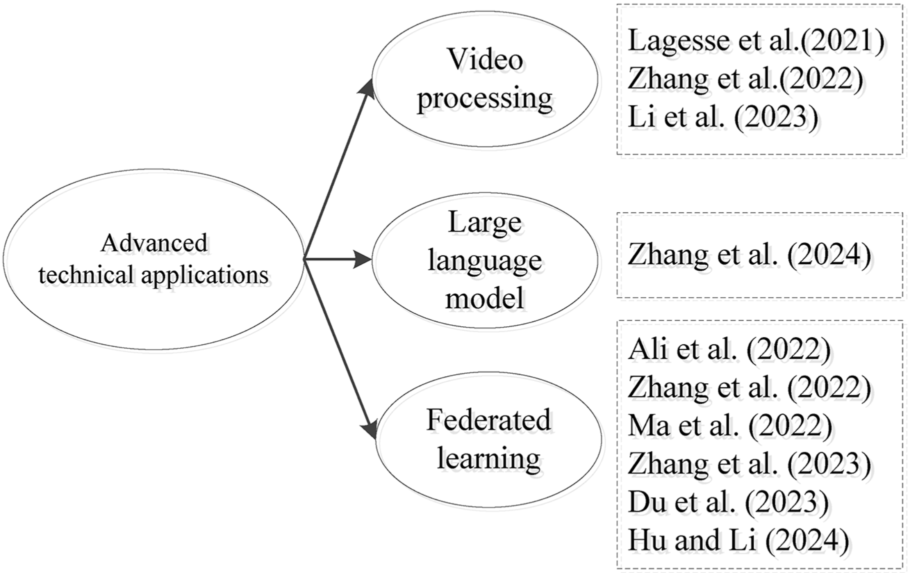

Fig. 6 Publications in the three advanced technical applications

rent video processing approaches based on approximate HE are discussed along with other techniques, such as the work in Li et al. (2023), while some schemes process video data alone (Lagesse et al. 2021; Zhang et al. 2022).

As smart devices with video recording capabilities continue to evolve, it has become easier to record the lives of individuals and the history of the world. However, some videos that record small groups of scenes may compromise privacy by revealing the presence of certain individuals. The work in Lagesse et al. (2021) addressesed this issue for privacy protection. The authors observed that Kullback-Leibler divergence, Bhattacharyya distance, and Cramer distance are more compatible with HE schemes and have lower errors than the four methods used to compute distances in Wu and Lagesse (2019). The scheme employs CKKS to encrypt video data for similarity computation on the ciphertext and has been implemented on various hardware devices, including Android, Raspberry Pi and PC. For privacy-preserving video similarity computation, this scheme achieves an accuracy of 99.32% and an F1 score of 97.97%. The work proposed by Zhang et al. (2022) investigates privacy-preserving Driver Drowsiness Detection. The scheme combines FL and Transfer Learning (TL) and utilizes the CKKS scheme. Then, the Privacy-Preserving Federated Transfer Learning for Driver Drowsiness Detection (PFTL-DDD) scheme is proposed. The scheme divides the video into images and processes them, making it an image processing scheme. Compared with the traditional FL scheme, this scheme freezes the parameters of the extraction layer and only exchanges the parameters of the classification layer. This results in less communication overhead, as shown in its theoretical analysis. Several experiments are conducted on two publicly available datasets for sleepiness detection, and the results demonstrate that this scheme outperforms traditional FL in terms of accuracy and efficiency. For instance, the experiment on the NTHU-DDD dataset shows that PFTL-DDD has an accuracy of 83.48%, whereas traditional FL only has 76.31%.

LLM represents an advanced text processing model based on extensive text data. Transformer-based LLMs have become exceedingly popular in AI following the release of ChatGPT by OpenAI on November 30, 2022, which has seen widespread adoption (Wu et al. 2023). The popularity of ChatGPT over the past two years has also brought significant privacy concerns, exemplified by incidents such as the internal code leak at Samsung Group (Gupta et al. 2023). When employing LLMs, particularly in non-real-time applications, safeguarding user data privacy and implementing privacy-preserving inference are crucial. Currently, there are studies dedicated to addressing these challenges (Dong et al. 2023; Zhang et al. 2024), including an approximate HE-based scheme (Zhang et al. 2024). LLM is generally implemented based on Transformer's deep network model. In Zhang et al. (2024), the authors implemented non-interactive secure inference on LLM for the first time, using RNS-CKKS (Cheon et al. 2019) to support more homomorphic multiplication operations and polynomial optimization techniques for different nonlinear activation functions and LN. For a tokenized input, it is processed by an Attention module and a feed-forward network module, each of which is followed by an LN layer and then is operated by an Argmax layer for the output of the final result. An Attention module needs to perform 3 matrix multiplications and 1 softmax function. A feed-forward network module requires 2 matrix multiplications and 1 GeLU function. Therefore, compared with other types of neural networks, the model based on Transformer requires more matrix multiplication operations and more polynomial approximations to handle the Softmax, GeLU, and Argmax functions as well as LN. For the exponential function in Softmax, the scheme uses the same method as

BumbleBee (Lu et al. 2023), i.e., Taylor expansion. In addition, the authors proposed two SIMD based parallel optimization techniques: SIMD compression and decompression, and SIMD slot folding which includes QuickSum and QuickMax, to optimize some homomorphic computations in the Transformer model. Experimental results show that the scheme requires 164 MB of bandwidth (368.6X reduction) to perform a secure inference in 1103 s (2.8X speedup) compared with the state-of-the-art scheme BOLT (Pang et al. 2024) which is an interactive scheme.

FL is a decentralized ML approach designed to train models without the need for a centralized dataset. In FL, a participant maintains its data locally and shares only the parameters of model updates, not the raw data. By training locally and protecting data privacy, FL enables participants to train a global model together, leading to better model performance and generalization capabilities. In a broad sense, FL has achieved privacy-preserving learning by training locally without directly letting the server know the local data. However, many studies have found that directly using FL for distributed learning still has the potential to leak local data. Therefore, from a narrow perspective, FL has to be coupled with other privacy-preserving techniques, such as HE, SMPC, etc., in order to realize privacy-preserving learning. In addition to the related work Ali et al. (2022) that has been presented in the technical application of text processing and the one Zhang et al. (2022) in video processing, there are many FL schemes that incorporate approximate HE (Ma et al. 2022; Zhang et al. 2023; Du et al. 2023; Hu and Li 2024).

FL requires multiple clients to assist the server in obtaining an aggregated model, typically employing traditional single-key or multi-key cryptosystems. The work Zhang et al. (2023) utilizes CKKS and an improved compressed sensing technique (Donoho 2006) to implement a communication efficient PPFL scheme. The datasets implemented in this scheme are four image datasets, such as MNIST. Therefore, it can also be classified in the technical field of image processing. The scheme uses the CKKS scheme to protect FL model training and implements two neural network models: ResNet and CNN. The authors executed a large number of experiments showing the advantages of this scheme in terms of communication and accuracy. Especially on the MNIST dataset, a better accuracy of 98.7% is obtained compared with the other four frameworks, with the accuracies of the other schemes ranging from 79.6% to 98.4%. In the work Hu and Li (2024), Hu and Li proposed the MASKCRYPT scheme, which applies an encryption mask to filter out a small portion of updates for encryption. This scheme confuses the training trajectory by maximizing the local loss value of exposed model weights, and then sends individual masks to a specialized mask consensus mechanism to obtain the final mask for all clients. Experimental results demonstrate that compared to encrypting the entire model update, MASKCRYPT can achieve a 4.15 times reduction in communication overhead at a lower encryption ratio, while still effectively protecting the clients' private data from inversion attacks.

Compared with the single-key CKKS scheme, the schemes with multi-key enjoy better security and can tolerate a certin number of users in FL. The work Ma et al. (2022) extends the MK-CKKS scheme by aggregating multiple public keys to obtain the final public key, and this design can be used to solve collusion attacks. This multi-key scheme is used to protect FL and implement CNNs at the model level. The experiments are conducted on the Jetson Nano IoT device platform (Kurniawan 2021) and the multimodal dataset UP-FALL (Martínez-Villaseñor et al. 2019) which contains features and image data. Therefore, the scheme can also be classified into the technical applications of feature processing and image processing. Experimental results show that compared with PPFL based on the Paillier and MK-CKKS schemes, this scheme maintains high accuracy while significantly reducing computational cost and consuming reasonable energy on end devices. For example, the scheme achieves an accuracy of 93.37% when the local epochs are $L = 20$, while the traditional FL is 93.80%, and the scheme only has an accuracy loss of 0.07%. The work Du et al. (2023) designs a multi-key CKKS scheme based on an SSR variant, i.e., Shamir's t-out-of-n linear secret sharing scheme (LSSS) (Shamir 1979). The proposed scheme which implements a t-user tolerant CKKS variant is used in an FL framework to implement privacy-preserving training of ResNet, CNN and RNN models. The datasets used for its implementation include Federated MNIST and Federated Shakespeare, thereby classifying it into the fields of both image and text processing. Additionally, experimental results demonstrate a reduction in communication overhead from 8.61 MB to 841.89 MB, and an improvement in computational overhead from 15.84x to 25.8x compared with traditional FL, Paillier-based FL, and Batchcrypt-based FL.

Since FL is currently a very hot research area, there are many similar PPFL schemes based on approximate HE, in addition to the related work presented above. To better understand the research in this area, refer to these related works Qiu et al. (2022), Imran et al. (2023), Hao et al. (2023), Fotohi et al. (2024), Nguyen et al. (2024).

4 Discussions

In this section, we present the application scenarios, models, and datasets of approximate HE based PPML, and then point out some future directions.

4.1 Application scenarios/models/datasets

For each of the four main technical applications, we sort out their application scenarios, the used models, and the common datasets.

4.1.1 Application scenarios

Fig. 7 shows the specific application scenarios of the four technical applications.

There are two specific application scenarios in feature processing, which are biomedicine and plant classification. The former is a popular practical area, including myocardial infarction detection, prostate cancer detection, etc. The latter is a more common application scenario in theory, which is based on the classification of the Iris dataset to validate a feature recognition algorithm.

The application scenarios in image processing primarily focus on two aspects: biometric recognition and general image classification. Biometric recognition is particularly prominent, especially during public health events where non-contact image recognition or authentication plays a crucial role. General image classification involves categorizing common datasets in image recognition, such as MNIST, to assess the efficacy of image processing algorithms.

There are a broader range of specific application scenarios in PPML schemes for text processing, including sentiment analysis on data sourced from popular platforms like Twit-

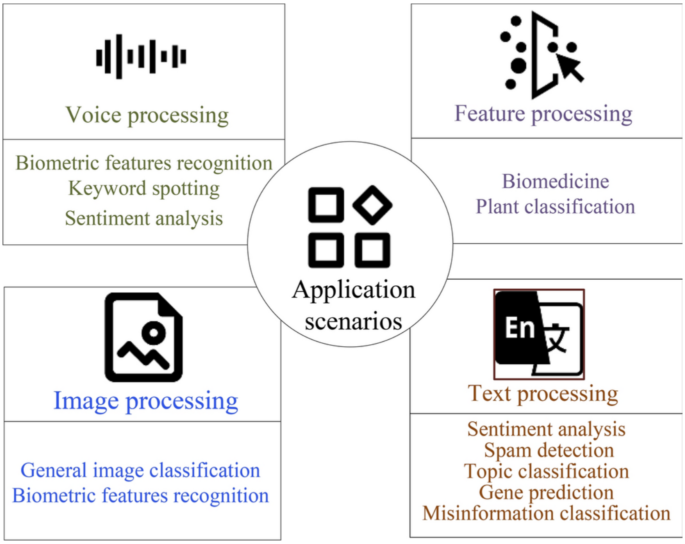

Fig. 7 The application scenarios used in four technical applications

ter and Amazon, spam detection on malicious emails, topic or misinformation classification, and gene prediction involving many-to-many predictions. With the rapid growth of information generation on the Internet, particularly driven by advances in generative AI such as ChatGPT, the need for privacy-preserving processing of these data has become increasingly urgent.

There are three application scenarios in PPML schemes for voice processing: biometric recognition, keyword recognition, and sentiment analysis. Voice, as an important biometric feature, can be used for contactless identification and authentication, like face recognition. Keyword recognition is a very popular field with the development of smart speakers and personal voice assistants. Waking up your smart speaker and personal voice assistant through a certain keyword and issuing some commands to them are very common scenarios in people's daily life nowadays. Sentiment analysis is a similar application area to sentiment analysis in text processing, and some of the related works involve converting the audio into text for recognition, while some researches directly process the audio to recognize text-independent sentiment features.

4.1.2 Models

Figure 8 shows the models used in each of the four technical applications.

The models for feature processing contain both traditional models and neural network models. The variety of traditional models that have been used for privacy protection with approximate HE is also very large, including logistic regression, KNN, decision trees, K-means, and SVMs. Others such as plain Bayesian models also deserve further research to develop them as privacy-preserving models. For the use of neural network models in privacy-preserving feature recognition, since this type of application is relatively simple, generally the simplest deep learning models can get very good accuracy, and in the current research work, the main use of shallow neural networks and extreme learning machines.

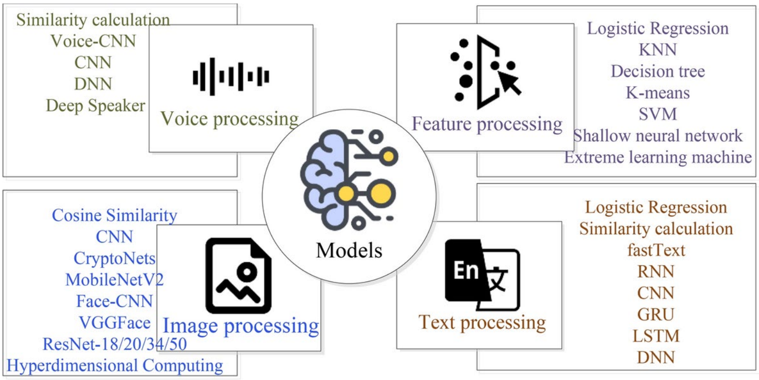

Fig. 8 The models used in four technical applications

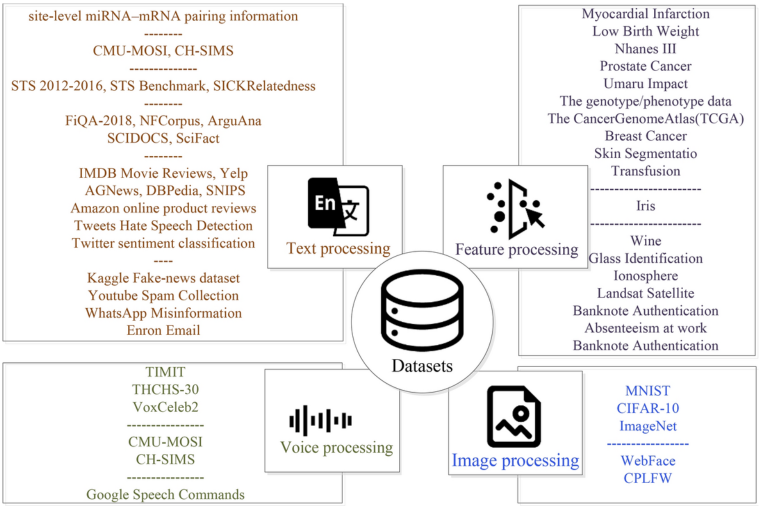

Fig. 9 The application scenarios of four technical applications

For text processing models also include traditional models and neural network models. But the former has only two models: logistic regression and similarity calculation. When using traditional models for text processing, there are not only models such as KNN (Srivastava et al. 2023) and decision tree (Jalal et al. 2022), which have already been combined with approximate HE schemes, but also plain Bayes (Ruan et al. 2022) and other models. All of these models have the potential to be further upgraded into privacy-preserving models. The deep network models in this area are much richer, and in addition to shallow neural networks, some special neural networks, such as CNN and RNN and some of their variant models, have been used to obtain better model performance.

For image processing the models are based on deep learning models except for similarity computation. These deep learning models range from basic CNNs to Face-CNN optimized for a particular domain, ResnNet with the introduction of residual function and MobileNet for mobile. In addition, the field is also designed to the introduction of privacy-preserving nature of CryptoNet and hyperdimensional computing models. Image recognition can be said to be the most widely used field with richer types of privacy-preserving models, and it is also the closest to life application scenarios. For example, in the field of mobile payment, privacy-preserving face recognition can be utilized for payment (Xu et al. 2018).

There are many current models for speech recognition, but the types of models that incorporate approximate HE are fewer, and the related literature is the smallest within the four domains. Except for privacy-preserving similarity computation for authentication, DNN, CNN and RNN models are dominant. And the Deep speaker used in it is the combined model of CNN and GRU.

4.1.3 Datasets

Feature-processed datasets are relatively simpler, typically comprising multiple columns of feature values corresponding to categorical labels. The datasets currently used for approximate HE based PPML can be categorized into three classes. The first class includes biomedical datasets such as disease datasets (e.g., Prostate Cancer), biometric feature datasets (e.g., Skin Segmentation), and gene datasets (e.g., TCGA). These datasets are often available from repositories like UCL [181], competition platforms like iDASH/Kaggle, and open-source platforms such as GitHub, or are included in various open source libraries of ML.

Text processing datasets generally involve natural language compositions, excluding those for gene prediction, such as site-level miRNA-mRNA pairing information and multimodal datasets (e.g., CMU-MOSI, CH-SIMS). Datasets used for approximate HE based PPML are categorized by task, including datasets for text similarity computation (e.g., STS 2012-2016, STS Benchmark, SICKRelatedness), text retrieval (e.g., FiQA-2018, NFCorpus, ArguAna, SCIDOCS, SciFact), and text classification. The latter encompasses datasets like IMDB Movie Reviews for sentiment classification, Kaggle Fake-news dataset for disinformation monitoring, and Enron Email for topic classification, which can also serve multi-class tasks like sentiment analysis.

Image processing datasets typically consist of matrices of pixel points with corresponding classification labels, varying in complexity from MNIST's 28x28 matrices to ImageNet's 224x224x3 matrices. Those used for PPML based on approximate HE are grouped into standard datasets for ML model testing (e.g., MNIST, CIFAR10, ImageNet, and their variants like F-MNIST) and datasets suitable for face recognition (e.g., Web Face, Cross-Pose LFW).

For audio processing datasets, applications include biometrics, keyword retrieval, and sentiment analysis, each with corresponding datasets such as TIMIT speech corpus for biometric recognition, Google Speech Commands for keyword retrieval, and CMU-MOSI for multimodal sentiment analysis.

4.1.4 Open source

In addition to the mentioned application areas, models, and datasets, another aspect that can assist researchers in further exploring this research area is the availability of the paper's code as open source. Several related works have already made their code open source, typi-

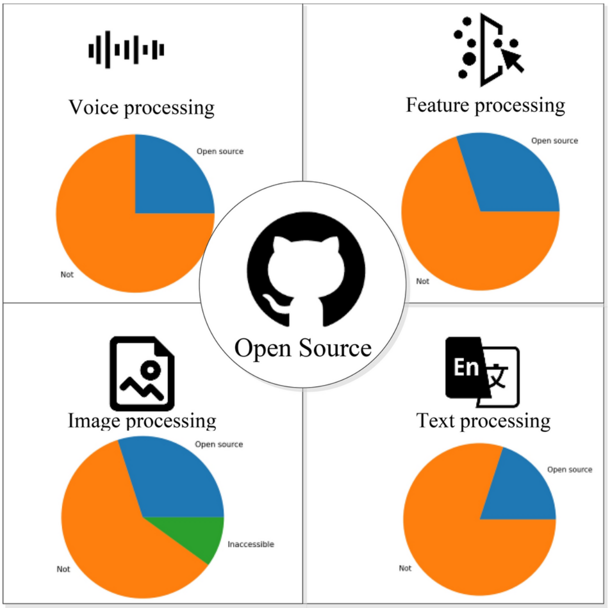

Fig. 10 The code public status in four technical applications

cally on GitHub. Figure 10 illustrates the open source status of related work across the four technical applications.

From Fig. 10, it is evident that among the four technical applications, the largest share of open source contributions lies in feature processing and image processing, each comprising 40%. However, in one instance of image processing, the provided URL is inaccessible, accounting for 10%. Voice processing follows with 25%, while text processing represents the smallest segment at 20%. The latter two areas require increased integration of open source code. Additionally, voice processing exhibits the fewest implementations of approximate HE, with only eight related works currently available.

4.2 Future directions

Currently, approximate HE has been deeply integrated with ML, and relevant schemes for different technical applications have emerged, providing better choices for users to protect their privacy. In order to better promote the development of PPML based on approximate HE, the following research directions deserve further depth.

- PPML based on approximate HE cannot progress without the respective advancements in both approximate HE and ML, as well as their mutual integration and collaborative improvement. Therefore, to better promote the development of PPML, it is necessary to optimize approximate HE schemes and ML models. In the area of approximate HE, we have already seen new techniques such as more efficient encoding scheme (Chen et al. 2024) and activation function computations within slots (Kim and Guyot 2023). Meanwhile, in the field of ML, new technologies continue to emerge, such as multimodal models (Liang et al. 2024) and generative adversarial networks (Brophy et al. 2023). Conducting in-depth research on these new directions will also advance the development of PPML.

- From the perspective of the four technical applications, it is the feature and image domains that currently have more related work, especially in the area of image recognition. In contrast, there is much less research work in audio and video. As the performance of personal devices, such as smartphones and bracelets, continues to improve, the application scenarios for audio and video continue to increase, and the scenarios that require the use of privacy-preserving privacy learning are becoming richer, such as voice assistants to operate cell phones.
- Model fusion is one of the future trends, especially in areas such as biometrics, where multi-feature inference can continuously improve accuracy and security, reduce bias and error, and provide multi-modal verification over single-feature inference. Therefore, there is a need to apply approximate HE to multimodal ML datasets and models to improve the breadth of applications of both HE and ML.
- Code disclosure plays a very important role in the development of a research field. For ML, privacy protection and PPML have been a lot of open source projects. However, for specific privacy protection schemes, especially some related works introduced in this work, open source is still a minority. The open source of these specific works can, on the one hand, better test and reproduce the schemes and also further promote the research in related fields.
- Secure training of larger neural networks and inference of very large scale neural networks are also crucial. For neural networks, we currently see some secure training schemes for shallow neural networks, and then some training schemes on deeper layers also in ResNet models, currently there is no special about approximate HE to realize the training of models like AlexNet or GoogLeNet. Secure inference on some large scale neural networks are currently implemented, for example ResNet-100. But there is no secure inference schemes for the deeper neural networks like AlexNet or GoogLeNet yet.
- One of the most dynamic areas in AI today is LLMs, as exemplified by ChatGPT. In the context of privacy-preserving inference for LLMs, numerous review articles have focused on LLMs and privacy security, primarily centered around safeguarding LLM parameters and accurately deriving LLMs (Yao et al. 2024; Das et al. 2024). However, there has been comparatively less effort in the LLM domain towards protecting user data privacy. Currently, only one study has employed approximate HE for privacy-preserving LLM inference, albeit limited to non-real-time applications (Zhang et al. 2024). Related research deserves to be further developed, especially the secure inference of LLMs for real-time applications.

5 Conclusion

This work reviews the development of ML and HE, leveraging the complementary advantages of approximate HE and ML to introduce the research topic of PPML based on approximate HE. The related literature is organized across four technical applications and three advanced applications. We examine the development status of PPML in these applications, along with commonly used datasets, models, and application scenarios. This organization aids researchers in understanding suitable ML datasets and models for different application scenarios and in identifying models that have been implemented for privacy-preserving training or inference. Additionally, this work concludes with a summary of related research and suggests future directions to support further advancement and widespread adoption of PPML based on approximate HE. Notably, there are many excellent works that focus on simultaneously using approximate HE along with SMPC or DP to enhance PPML schemes. By leveraging the strengths of multiple approaches, the performance of PPML can be significantly improved. In the future, we will conduct in-depth research in this area, aiming to achieve more robust and effective research outcomes.

Acknowledgements The study was supported by the Zhejiang Provincial Philosophy and Social Sciences Planning Project (grant/award number: 22NDJC188YB, 24NDQN120YBM)

Author contributions Jiangjun Yuan: Conceptualization, Writing-Original Draft, Formal Analysis, Investigation, Checking and Revising. Weinan Liu: Conceptualization, Methodology, Writing-Original Draft, Prepare Figures, Checking and Revising, Supervision, Theoretical Analysis. Jiawen Shi: Writing Original Draft, Prepare Figures, Checking and Revising, Supervision. Qingqing Li: Writing Original Draft, Formal analysis,Checking and Revising. All authors reviewed the manuscript.

Data availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare no Conflict of interest.

References

Michalski RS, Carbonell JG, Mitchell TM (2013) Machine Learning: An Artificial Intelligence Approach. Springer

Hsu C-Y, Lu Y-W (2023) Virtual metrology of material removal rate using a one-dimensional convolutional neural network-based bidirectional long short-term memory network with attention. Comp Ind Eng 186:109701

Vartiainen H, Tedre M, Valtonen T (2020) Learning machine learning with very young children: Who is teaching whom? Int J Child-Comp Inter 25:100182

Myszczynska MA, Ojamies PN, Lacoste AMB, Neil D, Saffari A, Mead R, Hautbergue GM, Holbrook JD, Ferraiuolo L (2020) Applications of machine learning to diagnosis and treatment of neurodegenerative diseases. Nat Rev Neurol 16(8):440–456

Rigaki M, García SR (2023) A survey of privacy attacks in machine learning. ACM Computing Surveys 56(4)

Isaak J, Hanna MJ (2018) User data privacy: facebook, cambridge analytica, and privacy protection. IEEE Comp Arch Lett 51(8):56–59

Zou Y, Roundy K, Tamersoy A, Shintre S, Roturier J, Schaub F (2020) Examining the adoption and abandonment of security, privacy, and identity theft protection practices. In: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, pp. 1–15

Baik JS (2020) Data privacy against innovation or against discrimination?: the case of the california consumer privacy act (ccpa). Telemat Inf 52:101431

Kok JW, Hoz MÁA, Jong Y, Brokke V, Elbers PW, Thoral P, Castillejo A, Trenor T, Castellano JM, Bronchalo AE et al (2023) A guide to sharing open healthcare data under the general data protection regulation. Sci Data 10(1):404

Hesamifard E, Takabi H, Ghasemi M, Wright RN (2018) Privacy-preserving machine learning as a service. Proceedings on Privacy Enhancing Technologies

Wang S, Zheng Y, Jia X (2023) Secgnn: Privacy-preserving graph neural network training and inference as a cloud service. IEEE Transactions on Services Computing

Wood A, Najarian K, Kahrobaei D (2020) Homomorphic encryption for machine learning in medicine and bioinformatics. ACM Comp Surv (CSUR) 53(4):1–35

Marcolla C, Sucasas V, Manzano M, Bassoli R, Fitzek FH, Aaraj N (2022) Survey on fully homomorphic encryption, theory, and applications. Procee IEEE 110(10):1572–1609

Sousa S, Kern R (2023) How to keep text private? a systematic review of deep learning methods for privacy-preserving natural language processing. Artif Intell Rev 56(2):1427–1492

Carbonell JG, Michalski RS, Mitchell TM (1983) An overview of machine learning. Machine learning, 3–23

Aguilar-Melchor C, Fau S, Fontaine C, Gogniat G, Sirdey R (2013) Recent advances in homomorphic encryption: a possible future for signal processing in the encrypted domain. IEEE Signal Proc Mag 30(2):108–117

Jordan MI, Mitchell TM (2015) Machine learning: trends, perspectives, and prospects. Science 349(6245):255–260

Martins P, Sousa L, Mariano A (2017) A survey on fully homomorphic encryption: an engineering perspective. ACM Comp Surv(CSUR) 50(6):1–33

Acar A, Aksu H, Uluagac AS, Conti M (2018) A survey on homomorphic encryption schemes: theory and implementation. ACM Comp Surv (Csur) 51(4):1–35

Alaya B, Laouamer L, Msilini N (2020) Homomorphic encryption systems statement: trends and challenges. Comp Sci Rev 36:100235

Liu B, Ding M, Shaham S, Rahayu W, Farokhi F, Lin Z (2021) When machine learning meets privacy: a survey and outlook. ACM Comp Surv (CSUR) 54(2):1–36

Murshed MS, Murphy C, Hou D, Khan N, Ananthanarayanan G, Hussain F (2021) Machine learning at the network edge: a survey. ACM Comp Surv (CSUR) 54(8):1–37

Munjal K, Bhatia R (2022) A systematic review of homomorphic encryption and its contributions in healthcare industry. Complex & Intelligent Systems, 1–28

Yao Z, Lum Y, Johnston A, Mejia-Mendoza LM, Zhou X, Wen Y, Aspuru-Guzik A, Sargent EH, Seh ZW (2023) Machine learning for a sustainable energy future. Nat Rev Mater 8(3):202–215

Kim M, Song Y, Wang S, Xia Y, Jiang X (2018) Secure logistic regression based on homomorphic encryption: design and evaluation. JMIR Med Inf 6(2):245–02550. https://doi.org/10.2196/medinform.8805

Lu W-j, Huang Z, Hong C, Ma Y, Qu H (2021) Pegasus: bridging polynomial and non-polynomial evaluations in homomorphic encryption. In: 2021 IEEE Symposium on Security and Privacy (SP). 2021 IEEE Symposium on Security and Privacy (SP), pp. 1057–1073. IEEE,

Koseki R, Ito A, Ueno R, Tibouchi M, Homma N (2023) Homomorphic encryption for stochastic computing. J Cryptogr Eng 13(2):251–263

Jiang X, Kim M, Lauter KE, Song Y (2018) Secure outsourced matrix computation and application to neural networks. Comp Sci 2018:1209–1222. https://doi.org/10.1145/3243734.3243837

Lee E, Lee J-W, Lee J, Kim Y-S, Kim Y, No J-S, Choi W (2022) Low-complexity deep convolutional neural networks on fully homomorphic encryption using multiplexed parallel convolutions. In: International Conference on Machine Learning, pp. 12403–12422. PMLR

Kim D, Guyot C (2023) Optimized privacy-preserving cnn inference with fully homomorphic encryption. IEEE Trans Inf Forens Secur 18:2175–2187

Rivest RL, Adleman L, Dertouzos ML et al (1978) On data banks and privacy homomorphisms. Found Secure Comp 4(11):169–180

Rivest RL, Shamir A, Adleman L (1978) A method for obtaining digital signatures and public-key cryptosystems. Commun ACM 21(2):120–126

Zhao S, Xu S, Han S, Ren S, Wang Y, Chen Z, Chen X, Lin J, Liu W (2023) Ppmm-da: Privacy-preserving multi-dimensional and multi-subset data aggregation with differential privacy for fog-based smart grids. IEEE Internet of Things Journal

Benaloh J (1994) Dense probabilistic encryption. In: Proceedings of the Workshop on Selected Areas of Cryptography, pp. 120–128

Paillier P (1999) Public-key cryptosystems based on composite degree residuosity classes. In: International Conference on the Theory and Applications of Cryptographic Techniques, pp. 223–238. Springer

He C, Liu G, Guo S, Yang Y (2022) Privacy-preserving and low-latency federated learning in edge computing. IEEE Int Things J 9(20):20149–20159

Han J, Yan L (2023) Adaptive batch homomorphic encryption for joint federated learning in cross-device scenarios. IEEE Internet of Things Journal

Boneh D, Goh E-J, Nissim K (2005) Evaluating 2-dnf formulas on ciphertexts. In: Theory of Cryptography: Second Theory of Cryptography Conference, TCC 2005, Cambridge, MA, USA, February 10-12, 2005. Proceedings 2, pp. 325–341. Springer

Gentry C (2009) A Fully Homomorphic Encryption Scheme. Stanford university

Brakerski Z, Gentry C, Vaikuntanathan V (2014) (Leveled) Fully Homomorphic Encryption Without Bootstrapping. ACM Trans Comp Theory (TOCT) 6(3):1–36

Fan J, Vercauteren F (2012) Somewhat practical fully homomorphic encryption. Cryptology ePrint Archive

Brakerski Z, Vaikuntanathan V (2014) Efficient fully homomorphic encryption from (standard) lwe. SIAM J Comp 43(2):831–871

Gentry C, Sahai A, Waters B (2013) Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In: Advances in Cryptology–CRYPTO 2013: 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I, pp. 75–92 . Springer

Ducas L, Micciancio D (2015) Fhew: bootstrapping homomorphic encryption in less than a second. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 617–640. Springer

Chillotti I, Gama N, Georgieva M, Izabachene M (2016) Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In: Advances in Cryptology–ASIACRYPT 2016: 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I 22, pp. 3–33. Springer

Regev O (2009) On lattices, learning with errors, random linear codes, and cryptography. J ACM (JACM) 56(6):1–40

Lyubashevsky V, Peikert C, Regev O (2010) On ideal lattices and learning with errors over rings. In: Advances in Cryptology–EUROCRYPT 2010, French Riviera, May 30–June 3, 2010. Proceedings 29, pp. 1–23. Springer

Cheon JH, Kim A, Kim M, Song Y (2017) Homomorphic encryption for arithmetic of approximate numbers. In: Advances in Cryptology–ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I 23, pp. 409–437. Springer

Jin W, Krishnamachari B, Naveed M, Ravi S, Sanou E, Wright K-L (2022) Secure publish-process-subscribe system for dispersed computing. In: 2022 41st International Symposium on Reliable Distributed Systems (SRDS), pp. 58–68. IEEE

Chen S, Zhao Y, Li R, Li X, Zhao J, Liu K (2022) Privacy preserving electronic scoring scheme based on ckks. In: 2022 IEEE 22nd International Conference on Communication Technology (ICCT), pp. 1884–1888. IEEE

Lai R, Fang X, Zheng P, Liu H, Lu W, Luo W (2022) Efficient fragile privacy-preserving audio watermarking using homomorphic encryption. In: International Conference on Artificial Intelligence and Security, pp. 373–385. Springer

Basuki A, Setiawan I, Rosiyadi D, Ramdhani TI, Susanto H (2022) Accelerating encrypted watermarking using wavelet transform and ckks homomorphic encryption. In: Proceedings of the 2022 International Conference on Computer, Control, Informatics and Its Applications, pp. 311–315

Albrecht MR, Player R, Scott S (2015) On the concrete hardness of learning with errors. J Math Cryptol 9(3):169–203

Guo Q, Nabokov D, Suvanto E, Johansson T (2024) Key recovery attacks on approximate homomorphic encryption with non-worst-case noise flooding countermeasures. In: Usenix Security

Cheon JH, Choe H, Kang M, Kim J (2024) Grafting: Complementing rns in ckks. Cryptology ePrint Archive

Al Badawi A, Bates J, Bergamaschi F, Cousins DB, Erabelli S, Genise N, Halevi S, Hunt H, Kim A, Lee Y et al (2022) Openfhe: Open-source fully homomorphic encryption library. In: Proceedings of the 10th Workshop on Encrypted Computing & Applied Homomorphic Cryptography, pp. 53–63

Microsoft SEAL (2023) (release 4.1). https://github.com/Microsoft/SEAL. Microsoft Research, Redmond, WA

Cheon JH, Han K, Kim A, Kim M, Song Y (2018) Bootstrapping for approximate homomorphic encryption. In: Advances in Cryptology–EUROCRYPT 2018: 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29-May 3, 2018 Proceedings, Part I 37, pp. 360–384. Springer

Cheon JH, Cho W, Kim J, Stehlé D (2023) Homomorphic multiple precision multiplication for ckks and reduced modulus consumption. In: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, pp. 696–710

Cheon JH, Han K, Kim A, Kim M, Song Y (2019) A full rns variant of approximate homomorphic encryption. In: Selected Areas in Cryptography–SAC 2018: 25th International Conference, Calgary, AB, Canada, August 15–17, 2018, Revised Selected Papers 25, pp. 347–368. Springer

Wang J, Yang C, Hou J, Zhang F, Meng Y, Su Y, Liu L (2024) A compact and efficient hardware accelerator for rns-ckks en/decoding and en/decryption. IEEE Transactions on Circuits and Systems II: Express Briefs (2024)

Chen H, Dai W, Kim M, Song Y (2019) Efficient multi-key homomorphic encryption with packed ciphertexts with application to oblivious neural network inference. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, pp. 395–412

Du W, Li M, Wu L, Han Y, Zhou T, Yang X (2023) A efficient and robust privacy-preserving framework for cross-device federated learning. Complex & Intelligent Systems, 1–15

Dwork C (2006) Differential privacy. In: International Colloquium on Automata, Languages, and Programming, pp. 1–12. Springer

Yao AC (1982) Protocols for secure computations. In: 23rd Annual Symposium on Foundations of Computer Science (sfcs 1982), pp. 160–164. IEEE

Mugunthan V, Polychroniadou A, Byrd D, Balch TH (2019) Smpai: Secure multi-party computation for federated learning. In: Proceedings of the NeurIPS 2019 Workshop on Robust AI in Financial Services, vol. 21. MIT Press Cambridge, MA, USA

Böhler J, Kerschbaum F (2021) Secure multi-party computation of differentially private heavy hitters. In: Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, pp. 2361–2377

Attrapadung N, Hanaoka G, Hiromasa R, Matsuda T, Schuldt JC (2023) Maliciously circuit-private multi-key fhe and mpc based on lwe. Des, Codes Cryptogr 91(5):1645–1684

Bian M, He G, Feng G, Zhang X, Ren Y (2023) Verifiable privacy-preserving heart rate estimation based on lstm. IEEE Internet of Things Journal

Dwork C, McSherry F, Nissim K, Smith A (2006) Calibrating noise to sensitivity in private data analysis. In: Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3, pp. 265–284. Springer

Dwork C, Rothblum GN, Vadhan S (2010) Boosting and differential privacy. In: 2010 IEEE 51st Annual Symposium on Foundations of Computer Science, pp. 51–60. IEEE

Erlingsson Ú, Pihur V, Korolova A (2014) Rappor: Randomized aggregatable privacy-preserving ordinal response. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, pp. 1054–1067

Dwork C, Rothblum GN (2016) Concentrated differential privacy. arXiv preprint arXiv:1603.01887

Cao Y, Yoshikawa M, Xiao Y, Xiong L (2018) Quantifying differential privacy in continuous data release under temporal correlations. IEEE Trans Knowl Data Eng 31(7):1281–1295

Tedeschi P, Al Nuaimi FA, Awad AI, Natalizio E (2023) Privacy-aware remote identification for unmanned aerial vehicles: current solutions, potential threats, and future directions. IEEE Trans Ind Inf 20(2):1069–1080

Mangold P, Perrot M, Bellet A, Tommasi M (2023) Differential privacy has bounded impact on fairness in classification. In: International Conference on Machine Learning, pp. 23681–23705. PMLR

Xu Z, Collins M, Wang Y, Panait L, Oh S, Augenstein S, Liu T, Schroff F, McMahan HB (2023) Learning to generate image embeddings with user-level differential privacy. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7969–7980

Guan J, Fang W, Huang M, Ying M (2023) Detecting violations of differential privacy for quantum algorithms. In: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, pp. 2277–2291

Shi Y, Yang Y, Wu Y (2024) Federated edge learning with differential privacy: An active reconfigurable intelligent surface approach. IEEE Transactions on Wireless Communications

Rabin MO (2005) How to exchange secrets with oblivious transfer. IACR Cryptol ePrint Arch 2005:187

Bellare M, Hoang VT, Rogaway P (2012) Foundations of garbled circuits. In: Proceedings of the 2012 ACM Conference on Computer and Communications Security, pp. 784–796

Shamir A (1979) How to share a secret. Commun ACM 22(11):612–613

Goldreich O, Micali S, Wigderson A (2019) How to play any mental game, or a completeness theorem for protocols with honest majority. In: Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali, pp. 307–328

Knott B, Venkataraman S, Hannun A, Sengupta S, Ibrahim M, Maaten L (2021) Crypten: secure multi-party computation meets machine learning. Adv Neural Inf Proc Syst 34:4961–4973

Li X, Dowsley R, De Cock M (2021) Privacy-preserving feature selection with secure multiparty computation. In: International Conference on Machine Learning, pp. 6326–6336. PMLR

Gao C, Yu J (2023) Securerc: a system for privacy-preserving relation classification using secure multi-party computation. Comp Sec 128:103142

Chen L, Xiao D, Yu Z, Zhang M (2024) Secure and efficient federated learning via novel multi-party computation and compressed sensing. Information Sciences, 120481

Gascón A, Schoppmann P, Balle B, Raykova M, Doerner J, Zahur S, Evans D (2016) Privacy-preserving distributed linear regression on high-dimensional data. Cryptology ePrint Archive

Juvekar C, Vaikuntanathan V, Chandrakasan A (2018) {GAZELLE}: A low latency framework for secure neural network inference. In: 27th USENIX Security Symposium (USENIX Security 18), pp. 1651–1669

Li S, Xue K, Zhu B, Ding C, Gao X, Wei D, Wan T (2020) Falcon: A fourier transform based approach for fast and secure convolutional neural network predictions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8705–8714

Mishra P, Lehmkuhl R, Srinivasan A, Zheng W, Popa RA (2020) Delphi: A cryptographic inference system for neural networks. In: Proceedings of the 2020 Workshop on Privacy-Preserving Machine Learning in Practice, pp. 27–30

Jha NK, Ghodsi Z, Garg S, Reagen B (2021) Deepreduce: Relu reduction for fast private inference. In: International Conference on Machine Learning, pp. 4839–4849. PMLR

Graepel T, Lauter K, Naehrig M (2012) Ml confidential: Machine learning on encrypted data. In: International Conference on Information Security and Cryptology, pp. 1–21. Springer

Choi H, Woo SS, Kim H (2024) Blind-touch: Homomorphic encryption-based distributed neural network inference for privacy-preserving fingerprint authentication. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, pp. 21976–21985

Hijazi NM, Aloqaily M, Guizani M, Ouni B, Karray F (2023) Secure federated learning with fully homomorphic encryption for iot communications. IEEE Internet of Things Journal

Li H, Wang T, Qiao Z, Yang B, Gong Y, Wang J, Qiu G (2021) Blockchain-based searchable encryption with efficient result verification and fair payment. J Inf Sec Appl 58:102791

Liu Z, Wan L, Guo J, Huang F, Feng X, Wang L, Ma J (2023) Ppru: A privacy-preserving reputation updating scheme for cloud-assisted vehicular networks. IEEE Transactions on Vehicular Technology

Pennington J, Socher R, Manning CD (2014) Glove: Global vectors for word representation. In: Conference on Empirical Methods in Natural Language Processing, pp. 1532–1543

Muhammad I, Yan Z (2015) Supervised machine learning approaches: A survey. ICTACT Journal on Soft Computing 5(3)

Kim A, Song Y, Kim M, Lee K, Cheon JH (2018) Logistic regression model training based on the approximate homomorphic encryption. BMC medical genomics 2018(Suppl 4),

Mihara K, Yamaguchi R, Mitsuishi M, Maruyama Y (2020) Neural network training with homomorphic encryption. Preprint at arXiv:2012.13552

Liu J, Wang C, Tu Z, Wang XA, Lin C, Li Z (2021) Secure knn classification scheme based on homomorphic encryption for cyberspace. Sec Commun Net. https://doi.org/10.1155/2021/8759922

T'Jonck K, Kancharla CR, Pang B, Hallez H, Boydens J (2022) Privacy preserving classification via machine learning model inference on homomorphic encrypted medical data. In: 2022 XXXI International Scientific Conference Electronics (ET). 2022 XXXI International Scientific Conference Electronics (ET), pp. 1–6. IEEE,

Li K, Huang R (2022) A ckks-based privacy preserving extreme learning machine. Int J Inf Sec 24(1):166–175

Hong S, Park JH, Cho W, Choe H, Cheon JH (2022) Secure tumor classification by shallow neural network using homomorphic encryption. BMC Genom 23(1):1–19. https://doi.org/10.1186/s12864-022-08469-w

Rovida L (2023) Fast but approximate homomorphic k-means based on masking technique. International Journal of Information Security, 1–15

Ypma TJ (1995) Historical development of the newton-raphson method. SIAM Rev 37(4):531–551

Bottou L (2010) Large-scale machine learning with stochastic gradient descent. In: Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers, pp. 177–186. Springer

Lu W-j, Zhou J-J, Sakuma J (2018) Non-interactive and output expressive private comparison from homomorphic encryption. In: Proceedings of the 2018 on Asia Conference on Computer and Communications Security, pp. 67–74

Tueno A, Boev Y, Kerschbaum F (2020) Non-interactive private decision tree evaluation. In: Data and Applications Security and Privacy XXXIV: 34th Annual IFIP WG 11.3 Conference, DBSec 2020, Regensburg, Germany, June 25–26, 2020, Proceedings 34, pp. 174–194. Springer

Goldschmidt RE (1964) Applications of division by convergence. PhD thesis, Massachusetts Institute of Technology

Wang W, Gan Y, Vong C-M, Chen C (2020) Homo-elm: fully homomorphic extreme learning machine. Int J Mach Learn Cybern 11:1531–1540

Kuri S, Hayashi T, Omori T, Ozawa S, Aono Y, Wang L, Moriai S et al (2017) Privacy preserving extreme learning machine using additively homomorphic encryption. In: 2017 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1–8. IEEE

Al Badawi A, Hoang L, Mun CF, Laine K, Aung KMM (2020) Privft: private and fast text classification with homomorphic encryption. IEEE Access 8:226544–226556

Podschwadt R, Takabi D (2020) Classification of encrypted word embeddings using recurrent neural networks. Web Search and Data Mining, 27–31

Podschwadt R, Takabi D (2021) Non-interactive privacy preserving recurrent neural network prediction with homomorphic encryption. IEEE International Conference on Cloud Computing, 65–70 https://doi.org/10.1109/CLOUD53861.2021.00019

Lee G, Kim M, Park JH, Hwang S-w, Cheon JH (2022) Privacy-preserving text classification on bert embeddings with homomorphic encryption. arXiv preprint arXiv:2210.02574

Kim D, Lee G, Oh S (2022) Toward privacy-preserving text embedding similarity with homomorphic encryption. In: Proceedings of the Fourth Workshop on Financial Technology and Natural Language Processing (FinNLP). Proceedings of the Fourth Workshop on Financial Technology and Natural Language Processing (FinNLP), pp. 25–36

Walch R, Sousa S, Helminger L, Lindstaedt S, Rechberger C, Trügler A (2022) Cryptotl: Private, efficient and secure transfer learning. arXiv preprint arXiv:2205.11935

Ali H, Tallal R, Qayyum A, Alghadhban A, Alazmi M, Alzamil A, AlUtaibi K, Qadir J (2022) Spam-das: Secure and privacy-aware misinformation detection as a service. TechRxiv, https://doi.org/10.36227/techrxiv.19351679.v1

Jang J, Lee Y, Kim A, Na B, Yhee D, Lee B, Cheon JH, Yoon S (2022) Privacy-preserving deep sequential model with matrix homomorphic encryption. In: Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security. ASIA CCS '22, pp. 377–391. Association for Computing Machinery, New York, NY, USA

Wang Z, Ikeda M (2023) High-throughput privacy-preserving gru network with homomorphic encryption. In: 2023 International Joint Conference on Neural Networks (IJCNN). 2023 International Joint Conference on Neural Networks (IJCNN), pp. 1–9

Li Z, Sang Y, Deng X, Tian H (2023) Lightweight and efficient privacy-preserving multimodal representation inference via fully homomorphic encryption. ACIIDS (1)

Panda S (2021) Principal component analysis using ckks homomorphic scheme. In: Cyber Security Cryptography and Machine Learning: 5th International Symposium, CSCML 2021, Be'er Sheva, Israel, July 8–9, 2021, Proceedings 5, pp. 52–70. Springer

Conneau A, Kiela D (2018) Senteval: An evaluation toolkit for universal sentence representations. arXiv preprint arXiv:1803.05449

Joulin A, Grave E, Bojanowski P, Mikolov T (2016) Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759

Fuzhen Z, Zhiyuan Q, Keyu D, Dongbo X, Yongchun Z, Hengshu Z, Hui X, Qing H (2021) A comprehensive survey on transfer learning. Procee IEEE 109(1):43–76

He Z, Zhang T, Lee RB (2019) Model inversion attacks against collaborative inference. In: Proceedings of the 35th Annual Computer Security Applications Conference, pp. 148–162

Abadi M, Chu A, Goodfellow I, McMahan HB, Mironov I, Talwar K, Zhang L (2016) Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 308–318

Driscoll TA, Hale N, Trefethen LN (2014) Chebfun guide. Pafnuty Publications, Oxford

Zhao Y, Komachi M, Kajiwara T, Chu C (2022) Region-attentive multimodal neural machine translation. Neurocomputing 476:1–13

Karpathy A, Fei-Fei L (2017) Deep visual-semantic alignments for generating image descriptions. IEEE Trans Pattern Analy Mach Intell 39(4):664–676

Devlin J, Chang M-W, Lee K, Toutanova K (2018) Bert: Pre-training of deep bidirectional transformers for language understanding. Preprint at arXiv:1810.04805

Boemer F, Costache A, Cammarota R, Wierzynski C (2019) ngraph-he2: A high-throughput framework for neural network inference on encrypted data. In: Proceedings of the 7th ACM Workshop on Encrypted Computing & Applied Homomorphic Cryptography, pp. 45–56

Ishiyama T, Suzuki T, Yamana H (2020) Highly accurate cnn inference using approximate activation functions over homomorphic encryption. In: 2020 IEEE International Conference on Big Data (Big Data). 2020 IEEE International Conference on Big Data (Big Data), pp. 3989–3995. IEEE, Jung W, Kim S, Ahn JH, Cheon JH, Lee Y (2021) Over 100x faster bootstrapping in fully homomorphic encryption through memory-centric optimization with gpus. IACR Transactions on Cryptographic Hardware and Embedded Systems, 114–148

Lee J-W, Kang H, Lee Y, Choi W, Eom J, Deryabin M, Lee E, Lee J, Yoo D, Kim Y-S (2022) Privacy-preserving machine learning with fully homomorphic encryption for deep neural network. IEEE Access 10:30039–30054

Lloret-Talavera G, Jorda M, Servat H, Boemer F, Chauhan C, Tomishima S, Shah NN, Peña AJ (2022) Enabling homomorphically encrypted inference for large dnn models. IEEE Trans Comp 71(5):1145–1155. https://doi.org/10.1109/TC.2021.3076123

Li L, Zhu H, Zheng Y, Wang F, Lu R, Li H (2022) Efficient and privacy-preserving fusion based multi-biometric recognition. Global Commun Conf. https://doi.org/10.1109/GLOBECOM48099.2022.10000971

Sperling L, Ratha N, Ross A, Boddeti VN (2022) Heft: homomorphically encrypted fusion of biometric templates. Int Conf Biom. https://doi.org/10.1109/IJCB54206.2022.10007995

Lee J-W, Lee E, Kim Y-S, No J-S (2023) Rotation key reduction for client-server systems of deep neural network on fully homomorphic encryption. In: International Conference on the Theory and Application of Cryptology and Information Security, pp. 36–68. Springer

Y, N, M, Z, S, G, G, DM, R, C, C, W, D, M, T R (2023) Efficient machine learning on encrypted data using hyperdimensional computing. In: 2023 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED). 2023 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED), pp. 1–6

Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L-C (2018) Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4510–4520

Gilad-Bachrach R, Dowlin N, Laine K, Lauter K, Naehrig M, Wernsing J (2016) Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In: International Conference on Machine Learning, pp. 201–210. PMLR

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778

Jang E, Gu S, Poole B (2016) Categorical reparameterization with gumbel-softmax. Preprint at arXiv:1611.01144

Nguyen K, Denman S, Sridharan S, Fookes C (2014) Score-level multibiometric fusion based on dempster-shafer theory incorporating uncertainty factors. IEEE Trans Human-Mach Syst 45(1):132–140

Ahmed D, Sabir A, Das A (2023) Spying through your voice assistants: realistic voice command fingerprinting. In: 32nd USENIX Security Symposium (USENIX Security 23), pp. 2419–2436

Deldjoo Y, Schedl M, Knees P (2024) Content-driven music recommendation: evolution, state of the art, and challenges. Comp Sci Rev 51:100618

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. Advances in neural information processing systems 30

Chindris M-C, Togan M, Arseni S-C (2020) Secure speaker recognition system using homomorphic encryption. Sec Inf Technol Commun. https://doi.org/10.1007/978-3-030-69255-1_13

Rahulamathavan Y (2022) Privacy-preserving similarity calculation of speaker features using fully homomorphic encryption. Preprint at arXiv:2202.07994

Zheng P, Cai Z, Zeng H, Huang J (2022) Keyword spotting in the homomorphic encrypted domain using deep complex-valued cnn. ACM Int Conf Multimed. https://doi.org/10.1145/3503161.3548350

Elworth DL, Kim S (2022) Hekws: Privacy-preserving convolutional neural network-based keyword spotting with a ciphertext packing technique. 2022 IEEE 24th International Workshop on Multimedia Signal Processing (MMSP), 01–06 https://doi.org/10.1109/MMSP55362.2022.9949982

Zhang Q-y, Wen Y-w, Huang Y-b, Li F-p (2024) Secure speech retrieval method using deep hashing and ckks fully homomorphic encryption. Multimedia Tools and Applications

Tang R, Lin J (2017) Honk: A pytorch reimplementation of convolutional neural networks for keyword spotting. Computing Research Repository arXiv:1710.06554

Lagesse B, Nguyen G, Goswami U, Wu K (2021) You had to be there: Private video sharing for mobile phones using fully homomorphic encryption. In: 2021 IEEE International Conference on Pervasive Computing and Communications Workshops and Other Affiliated Events (PerCom Workshops), pp. 730–735. IEEE

Zhang L, Saito H, Yang L, Wu J (2022) Privacy-preserving federated transfer learning for driver drowsiness detection. IEEE Access 10:80565–80574

Wu K, Lagesse B (2019) Do you see what i see?< subtitle> detecting hidden streaming cameras through similarity of simultaneous observation. In: 2019 IEEE International Conference on Pervasive Computing and Communications (PerCom, pp. 1–10. IEEE

Wu T, He S, Liu J, Sun S, Liu K, Han Q-L, Tang Y (2023) A brief overview of chatgpt: the history, status quo and potential future development. IEEE/CAA J Autom Sinica 10(5):1122–1136

Gupta M, Akiri C, Aryal K, Parker E, Praharaj L (2023) From chatgpt to threatgpt: impact of generative ai in cybersecurity and privacy. IEEE Access 11:80218–80245. https://doi.org/10.1109/ACCESS.2023.3300381

Dong Y, Lu W-j, Zheng Y, Wu H, Zhao D, Tan J, Huang Z, Hong C, Wei T, Chen W (2023) PUMA: Secure Inference of LLaMA-7B in Five Minutes

Zhang J, Liu J, Yang X, Wang Y, Chen K, Hou X, Ren K, Yang X (2024) Secure transformer inference made non-interactive. Cryptology ePrint Archive

Lu W-j, Huang Z, Gu Z, Li J, Liu J, Ren K, Hong C, Wei T, Chen W (2023) Bumblebee: Secure two-party inference framework for large transformers. Cryptology ePrint Archive

Pang Q, Zhu J, Möllering H, Zheng W, Schneider T (2024) Bolt: Privacy-preserving, accurate and efficient inference for transformers. In: 2024 IEEE Symposium on Security and Privacy (SP), pp. 130–130. IEEE Computer Society

Ma J, Naas S-A, Sigg S, Lyu X (2022) Privacy-preserving federated learning based on multi-key homomorphic encryption. Int J Intell Syst 37(9):5880–5901

Zhang Y, Miao Y, Li X, Wei L, Liu Z, Choo K-KR, Deng RH (2023) Efficient privacy-preserving federated learning with improved compressed sensing. IEEE Transactions on Industrial Informatics

Hu C, Li B (2024) Maskcrypt: Federated learning with selective homomorphic encryption. IEEE Transactions on Dependable and Secure Computing

Donoho DL (2006) Compressed sensing. IEEE Trans Inf Theory 52(4):1289–1306

Kurniawan A (2021) Iot projects with nvidia jetson nano. IoT Projects with NVIDIA Jetson Nano

Martínez-Villaseñor L, Ponce H, Brieva J, Moya-Albor E, Núñez-Martínez J, Peñafort-Asturiano C (2019) Up-fall detection dataset: a multimodal approach. Sensors 19(9):1988

Qiu F, Yang H, Zhou L, Ma C, Fang L (2022) Privacy preserving federated learning using ckks homomorphic encryption. In: International Conference on Wireless Algorithms, Systems, and Applications, pp. 427–440. Springer

Imran M, Yin H, Chen T, Nguyen QVH, Zhou A, Zheng K (2023) Refrs: resource-efficient federated recommender system for dynamic and diversified user preferences. ACM Trans Inf Syst 41(3):1–30

Hao X, Lin C, Dong W, Huang X, Xiong H (2023) Robust and secure federated learning against hybrid attacks: a generic architecture. IEEE Transactions on Information Forensics and Security

Fotohi R, Aliee FS, Farahani B (2024) A lightweight and secure deep learning model for privacy-preserving federated learning in intelligent enterprises. IEEE Internet of Things Journal

Nguyen C-H, Saputra YM, Hoang DT, Nguyen DN, Nguyen V-D, Xiao Y, Dutkiewicz E (2024) Encrypted data caching and learning framework for robust federated learning-based mobile edge computing. IEEE/ACM Transactions on Networking

Srivastava SK, Vidyarthi A, Singh SK (2023) Modified ml-knn: Role of similarity measures and nearest neighbor configuration in multi-label text classification on big social network graph data. In: Advances in Computers vol. 128, pp. 287–312. Elsevier,

Jalal N, Mehmood A, Choi GS, Ashraf I (2022) A novel improved random forest for text classification using feature ranking and optimal number of trees. J King Saud Univ-Comp Inf Sci 34(6):2733–2742

Ruan S, Chen B, Song K, Li H (2022) Weighted naïve bayes text classification algorithm based on improved distance correlation coefficient. Neural Computing and Applications, 1–10

Xu W, Shen Y, Bergmann N, Hu W (2018) Sensor-assisted multi-view face recognition system on smart glass. IEEE Transactions on Mobile Computing 17(1)

UCL Research Data Repository. http://archive.ics.uci.edu/datasets

Chen D, Qu H, Xu G (2024) AegisFL: Efficient and flexible privacy-preserving byzantine-robust cross-silo federated learning. In: Forty-first International Conference on Machine Learning. https://openreview.net/forum?id=PHUAG63Efe

Liang PP, Zadeh A, Morency L-P (2024) Foundations & trends in multimodal machine learning: principles, challenges, and open questions. ACM Comp Surv 56(10):1–42

Brophy E, Wang Z, She Q, Ward T (2023) Generative adversarial networks in time series: a systematic literature review. ACM Comp Surv 55(10):1–31

Yao Y, Duan J, Xu K, Cai Y, Sun Z, Zhang Y (2024) A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. High-Confidence Computing 4(2)

Das BC, Amini MH, Wu Y (2024) Security and privacy challenges of large language models: A survey. Preprint at arXiv:2402.00888

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.