Journal of Imaging Informatics in Medicine
https://doi.org/10.1007/s10278-024-01384-4

ORIGINAL PAPER

A Faster Privacy-Preserving Medical Image Diagnosis Scheme with Machine Learning

Jiuhong Ran[1] · **Dong Li**[2]

Received: 9 September 2024 / Revised: 21 November 2024 / Accepted: 12 December 2024

Abstract
Convolutional neural networks (CNNs) have become indispensable to medical image diagnosis research, enabling the automated differentiation of diseased images from extensive medical image datasets. Due to their efficacy, these methods raise significant privacy concerns regarding patient images and diagnostic models. To address these issues, some researchers have explored privacy-preserving medical image diagnosis schemes using fully homomorphic encryption (FHE). However, these schemes often support and are suitable for only a limited number of non-linear layers, resulting in less effective diagnoses and potentially inaccurate results. To improve upon these limitations, we propose and design a robust privacy-preserving medical diagnosis scheme that maintains both diagnostic accuracy and effectiveness at the same time. First, we utilize FHE to encrypt both the image and the model to safeguard the confidentiality of medical data and the model itself. Then, we introduce batch normalization to facilitate the use of multiple non-linear layers in deep convolutional neural networks within a ciphertext context. Furthermore, we employ a 2-degree polynomial function to approximate the ReLU activation function effectively. Finally, we introduce two innovative network depth optimization techniques to solve the issue of CNN depth insufficiency. Both theoretical and empirical analyses confirm that our scheme not only protects the confidentiality of medical images and diagnostic models but also ensures practicality and efficiency.

Keywords CNN · FHE · Batch normalization · Image diagnosis

Introduction

Amidst the swift progress of artificial intelligence (AI), AI-driven digital medical image diagnosis has become a pivotal component of mobile health diagnostics. Recently, the most advanced medical image diagnosis methods have increasingly utilized convolutional neural networks (CNNs) [1–6]. These networks offer significantly faster and more accurate diagnoses compared to traditional technologies such as support vector machines (SVMs) [7–9] and k-means clustering algorithms [10–12]. The framework for medical image diagnosis using CNNs is illustrated in Fig. 1. For instance, consider a patient with a pneumonia-related medical image who seeks a diagnosis. To obtain accurate diagnostic results, the patient uploads their medical image to the cloud-based diagnosis model. The model then processes the image and returns the diagnostic results to the patient. This process ensures that the patient receives timely and precise information about their health condition directly.

However, the appearance of digital medical imaging and automated diagnosis systems incurs significant challenges related closely to the privacy and security of sensitive medical data. The primary concern in utilizing CNNs for medical image classification lies in the handling and protection of patient-specific data, which often contains highly sensitive personal information. Medical images, by their nature, are classified as protected health information under regulations like HIPAA in the United States, GDPR in Europe, and similar laws worldwide. How to ensure the confidentiality and integrity of this data throughout its lifecycle from collection and analysis to storage and disposal is paramount.

Jiuhong Ran and Dong Li contributed equally to this work.

✉ Dong Li
lilvmy@163.com

Jiuhong Ran
1422596478@qq.com

[1] Hospital of Chongqing University, Chongqing University, No. 174, Shazheng Street, Shapingba, 400044 Chongqing, China

[2] College of Computer, Chongqing University, No. 55 Daxuecheng South Rd, Shapingba, 401331 Chongqing, China

Published online: 03 January 2025

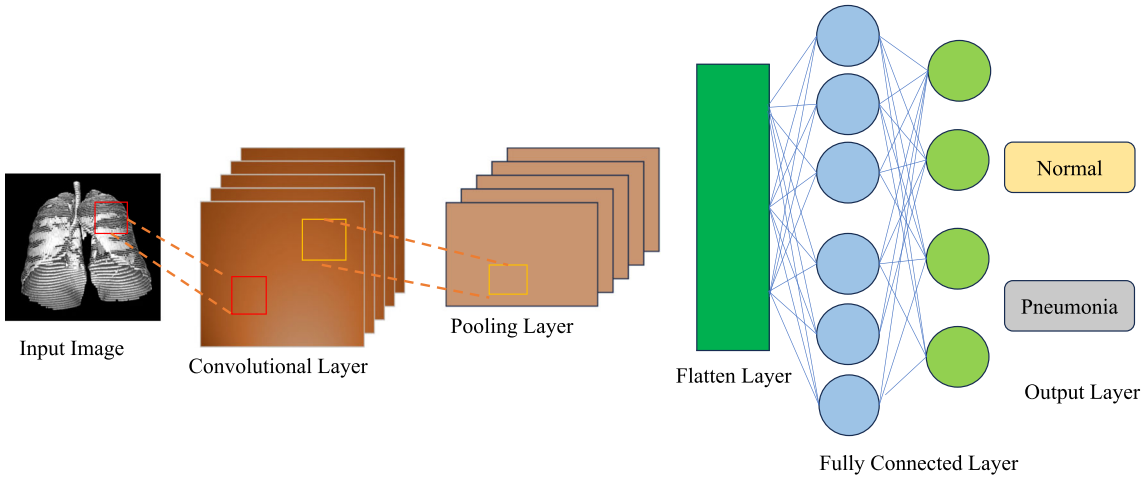


Fig. 1 The architecture of medical image diagnosis with CNN

Moreover, the parameters of the CNN models themselves can inadvertently become sources of data leakage. These models, when improperly secured, can be reverse-engineered to reveal information about the training data, potentially exposing patient data embedded within. This risk is heightened by the increasing trend of utilizing cloud-based platforms for computational support in processing large datasets, where the control over data security is not always transparent or guaranteed.

To guarantee the confidentiality of the patient's medical image and the parameters of the medical diagnosis model, researchers have begun to solve this issue from several perspectives, which include scientific [13], technological [14, 15], and legislative [16]. We have given the comparison of medical image diagnosis schemes with different privacy-preserving technologies in Table 1. From Table 1, it becomes evident that among the various privacy-preserving technologies, FHE [17–20] emerges as the optimal solution for ensuring the confidentiality of medical images and diagnostic model parameters. FHE stands out mainly because it can execute operations directly on encrypted data, ensuring data privacy remains intact during the analysis phase. This capability is especially advantageous in healthcare, where the sensitivity of data is critical, and maintaining privacy must not hinder the precision of diagnostic outcomes. Furthermore, FHE offers a robust security model which is not dependent on less secure, but traditional methods of encryption that require decryption during processing, thereby eliminating potential vulnerabilities. Another significant advantage is that FHE allows for complex operations, including those required by deep learning models such as CNNs, to be executed without exposing the underlying data. This capability is crucial for leveraging advanced AI techniques in medical diagnosis while strictly adhering to privacy regulations. Overall, while other technologies like secure multi-party computation and differential privacy also offer valuable benefits, FHE's comprehensive security and functionality make it the superior choice for privacy-preserving medical image diagnosis.

Unfortunately, using fully homomorphic encryption (FHE) in medical image diagnosis with CNN presents two limitations: (1) FHE substantially increases computational complexity, resulting in slower processing times; (2) the restricted depth of CNN and suboptimal methods for approximating non-linear functions, such as the use of quadratic polynomial function to approximate the ReLU functions, can compromise accuracy and diminish the effectiveness of learning and prediction in ciphertext diagnosis environment.

To address these limitations, we propose and design a robust privacy-preserving medical image diagnosis scheme in this paper. Specifically, we utilize the CKKS homomorphic encryption scheme [21] to simultaneously encrypt both the medical images and the parameters of the medical image diagnosis model, ensuring the confidentiality of these

Table 1 Comparative evaluation of privacy-enhancing technologies in medical image diagnostics

Privacy-preserving technologies	Secure data processing	Single-round processing
Fully homomorphic encryption	Yes	Yes
Secure multi-party computation	Yes	No
Croup-based anonymity	No	Yes
Differential privacy	No	Yes

sensitive data. The choice of CKKS over other FHE schemes such as BGV or BFV was primarily influenced by its inherent capability to handle float number arithmetic directly. This feature is crucial for applications involving medical images where pixel values are typically represented as float numbers. Using integer-only FHE schemes (e.g., BGV, BFV) would require quantization of these values into integers, potentially leading to significant information loss and consequently lower accuracy in classification results. In addition, CKKS allows dynamic adjustment of security parameters to balance security and performance. Moreover, we develop a 2-degree polynomial function to approximate the ReLU function and incorporate batch normalization within the ciphertext diagnosis environment to enhance the accuracy of the medical image diagnosis model. In addition, we design two neural network layer optimization strategies: integrating the convolutional layer with the batch normalization layer and merging the convolutional layer with the average pooling layer. These strategies not only increase the depth of the medical image model but also reduce the number of ciphertext multiplications, thereby accelerating the diagnosis process in the ciphertext environment. These enhancements aim to significantly improve both the performance and the security of medical image diagnosis. The main contributions of our scheme are as follows.

- We use the CKKS scheme to encrypt the medical images and the parameters of the medical diagnosis model aim to guarantee the confidentiality of these data. In addition, the CKKS scheme supports Single Instruction Multiple Data (SIMD) process operation, which can perform the same operation on multiple pieces of data simultaneously, significantly enhancing ciphertext diagnosis efficiency.
- To improve the accuracy of the medical image diagnosis model, we develop a 2-degree polynomial function to approximate the non-linear ReLU function and incorporate batch normalization within the ciphertext diagnosis environment, where the advantage of using batch normalization can increase the number of non-linear layers of the medical image diagnosis model.
- We design two neural network layer optimization strategies that not only increase the depth of the medical image model but also reduce the number of ciphertext multiplications required. Moreover, our experimental analysis further demonstrates the robustness and efficiency of our scheme, highlighting its potential for practical implementation in secure medical image processing environments.

The remainder of this paper is structured as follows: "Literature Survey" provides a review of related work on image classification using homomorphic encryption. "Methodology" details the construction of our proposed scheme. "Performance Analysis" demonstrates the robustness and efficiency of our approach through experimental analysis. Finally, "Conclusion" summarizes the key findings and contributions of this paper.

Literature Survey

The concept of using homomorphic encryption (HE) to ensure data privacy during computations had first been introduced in [22], where privacy homomorphisms were defined as encryption functions that allowed for operations on the encrypted data without the need for decryption. Early HE schemes had been limited to supporting either additions [23–25] or multiplications [26]. The pioneering HE scheme that supported both operations had been introduced in [27], utilizing ideal lattice-based cryptography to manage both additions and multiplications securely. Subsequently, the work in [28] had deviated from the ideal lattice approach, opting instead for integer polynomial rings for ciphertexts, which slightly reduced security constraints. The Brakerski-Gentry-Vaikuntanathan (BGV) scheme [29] had later modified this, using polynomial rings for ciphertexts and leveraging the learning with errors (LWE) and ring learning with errors (RLWE) challenges to ensure robustness and security.

Cryptonets [17] represents the pioneering approach to secure image classification, employing the YASHE [30] homomorphic encryption scheme to encrypt both the input image and the parameters of the pre-trained model. It also utilizes the square function as a substitute for the ReLU function to accommodate the encryption constraints. The only distinction of BFVML [20] lies in its use of the Brakerski/Fan-Vercauteren (BFV) [31] homomorphic encryption scheme for encrypting and encoding its data and model. Nevertheless, both schemes demand substantial computational resources, which hinders their practical deployment in real-life applications. To improve the inference efficiency, the works [33, 34] had provided a rapid HE solution for discretized CNN inference, and the nGraph-HE framework [35] had enabled the training of CNNs in plaintext on specialized hardware, followed by the deployment of these models onto HE systems processing the encrypted data, simplifying HE integration while adhering to an as-a-service model. Furthermore, Mishra [36] introduced a strategy that utilizes online and offline flags to address the challenge of users needing to remain online for extended durations.

Moreover, initiatives like [37] had explored secure multiparty computation (SMC), allowing multiple parties to collaboratively compute a function while each retained only partial data knowledge. Although these efforts had not integrated HE, the works [32, 38–40] had combined SMC with Paillier HE [25] for CNN applications, ensuring the privacy of both data and CNN models during computations.

For example, MiniONN [32] proposed a secure image classification framework that can transform any common neural network into an oblivious form and aims to achieve significantly lower inference latency. Furthermore, frameworks such as Gazelle [41] had integrated SMC and HE to support low-latency inference for CNNs, further enhancing the security and efficiency of these computational models.

However, the aforementioned secure image classification schemes based on FHE are limited to supporting only integer pixel values, whereas actual image pixels are typically floating point numbers. Consequently, these schemes must truncate the image pixels to comply with FHE requirements, inevitably leading to data distortion and reduced classification accuracy. Moreover, schemes reliant on SMC and semi-FHE incur prohibitive communication costs due to the need for multiple interactions between the client and the server, rendering them impractical for real-world applications. In contrast, our approach leverages the capabilities of CKKS homomorphic encryption, which supports floating point calculations and eliminates the need to truncate images, thereby avoiding data distortion and enhancing image classification accuracy. Additionally, we introduce two neural network optimization techniques aimed at boosting classification efficiency.

Methodology

CKKS Homomorphic Encryption

In this section, we provide a concise overview of the CKKS homomorphic encryption technology employed in our scheme.

CKKS [21] homomorphic encryption technology provides an approximate algorithm for encrypting real numbers, in which the plaintext and the ciphertext spaces are constructed based on the structure of the integer polynomial ring, where the security of the CKKS depends on RLWE problem. However, since sample data often appears in vector form, it needs to be encoded into polynomial form. The algorithm introduces a modest amount of noise during the encryption phase to obscure the message. This noise budget is set at the scheme's inception, and each operation on the ciphertext depletes this initial allocation. When the noise budget is reached, any attempt to decrypt will yield incorrect outcomes. In the CKKS framework, fixed-point values within real or complex vector data are processed using Single Instruction Multiple Data (SIMD) technology, translating these numbers into the realm of complex space for further operations. The transformation process involves encoding these complex vectors into polynomial forms, but this transition incurs a loss in precision. To mitigate this effect, the initial plaintext vector is amplified by a scaling factor, which helps preserve an acceptable level of precision in subsequent encodings. Owing to its proficiency with real numbers, it finds frequent application in the domains of data science and machine learning.

Figure 2 shows the encryption procedure of the CKKS encryption scheme, the value of N is the power of two, $\mathbb{C}^{N/2}$ is a complex space, $\mathbb{R} = \mathbb{Z}[X]/(X^N + 1)$ denotes a integer ring, and X^N+1 is a cyclotomic polynomial. Given fixed base q, the scheme chose a Residue Number System (RNS) base, composed of prime numbers $(q_0, \cdots, q_L)$, each roughly equivalent in magnitude to q. For $\forall l \in [0, L]$, the ciphertext exists as pair of polynomials within $\mathbb{R}_Q = \mathbb{R}/(Q_l \cdot \mathbb{R})$, where $Q_l = \prod_{i=0}^{l} q_i$. During the rescaling phase, RNS applies rescaling techniques to transition the encrypted plaintext message m from a level l to a level of $q_l^{-1} \cdot m$ to $l-1$, aiming to maintain nearly equivalent encryption precision across levels. Furthermore, the CKKS encryption scheme leveraged the RNS and the Number Theoretic Transformation (NTT) to enhance its framework. The processing of multiple inverses, as well as exponential or sigmoid functions, required only 160 ms for inputs with 32-bit precision across 213 slots, averaging 20 ms per slot. For statistical computations, determining the mean and variance of 213 real values took 307 ms and 518 ms, respectively.

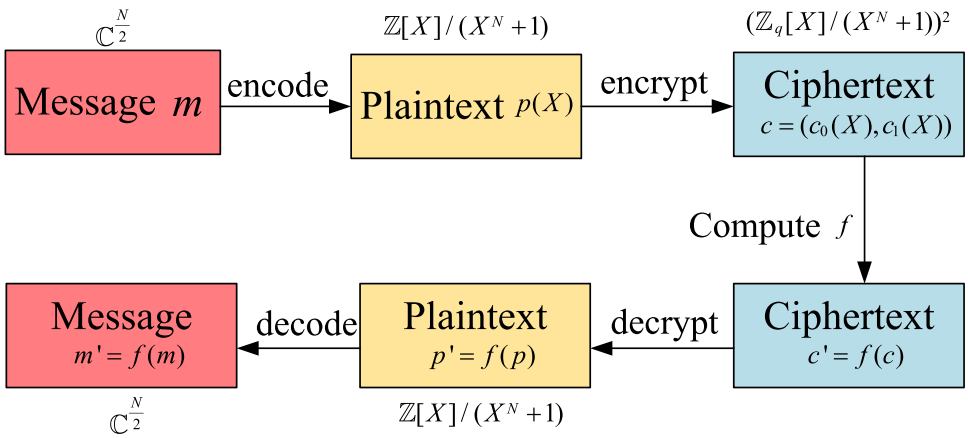


Fig. 2 CKKS encryption procedure

CNN

At a conceptual level, a neural network is organized into neurons that are grouped into layers in sequence. Each neuron processes an input through a specific function and produces a result. Beyond the initial input layer and the final output layer, neural networks typically include one or more intermediate layers, referred to as hidden layers. In fully feed-forward architectures, each neuron connects to every neuron in the next layer via weighted connections. Neurons differ across layers in their functionality. For example, neurons in the input layer take a single input and their output directly reflects this input. In contrast, neurons in hidden layers handle multiple inputs, calculate their weighted aggregate, apply an activation function to this aggregate, and then produce an output based on this function. Common types of these activation functions include Sigmoid, Max, or Mean.

As a specific type of feed-forward neural network, CNN is widely used in the fields of image recognition and classification due to its numerous advantages. The main architecture of CNN is as follows.

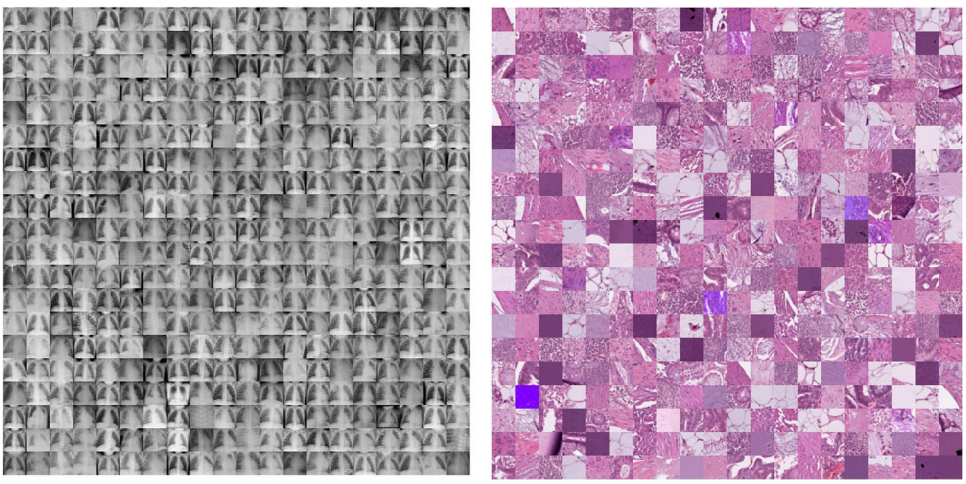
- *Convolutional layer*: These layers apply a set of learnable filters (kernels) to the input image to produce feature maps, where the filters are $n \times n$-dimensional matrices with a stride. The convolution operation captures local dependencies by moving the filters across the image and computing dot products between the filter values and the receptive field's pixel values, which allows CNNs to learn spatial features such as edges, textures, and patterns.
- *Activation layer*: Following each convolutional layer, an activation function is applied to introduce non-linearity into the model. The rectified linear unit (ReLU) is commonly used for this purpose, denoted as $f(x) = \max(0, x)$.
- *Pooling layer*: Also known as sub-sampling or down-sampling layers, pooling layers reduce the dimensionality of feature maps while retaining the most important information. Max pooling and average pooling are widely used techniques, where max pooling selects the maximum value from each pooling window, and average pooling computes the average.
- *Fully connected layer*: These layers are similar to those in traditional neural networks, where each neuron is connected to every neuron in the previous layer. Fully connected layers are typically placed towards the end of the network and are used to combine the features learned by convolutional and pooling layers for final classification or regression tasks.

Dataset

We utilize two types of medical images from the MedMNIST data repository: the PneumoniaMNIST dataset and the PathMNIST dataset [42]. The PneumoniaMNIST dataset comprises 5656 chest X-ray images, enabling binary classification of pneumonia versus normal cases. It includes 4708 training images and 624 test images, each with dimensions of $1 \times 28 \times 28$ pixels. The PathMNIST dataset contains 107,180 colon pathology images for predicting survival from colorectal cancer histology, with 89,996 training images and 7180 test images, also sized at $1 \times 28 \times 28$ pixels. Examples of images from the PneumoniaMNIST dataset and the PathMNIST dataset are shown in Fig. 3.

System Model

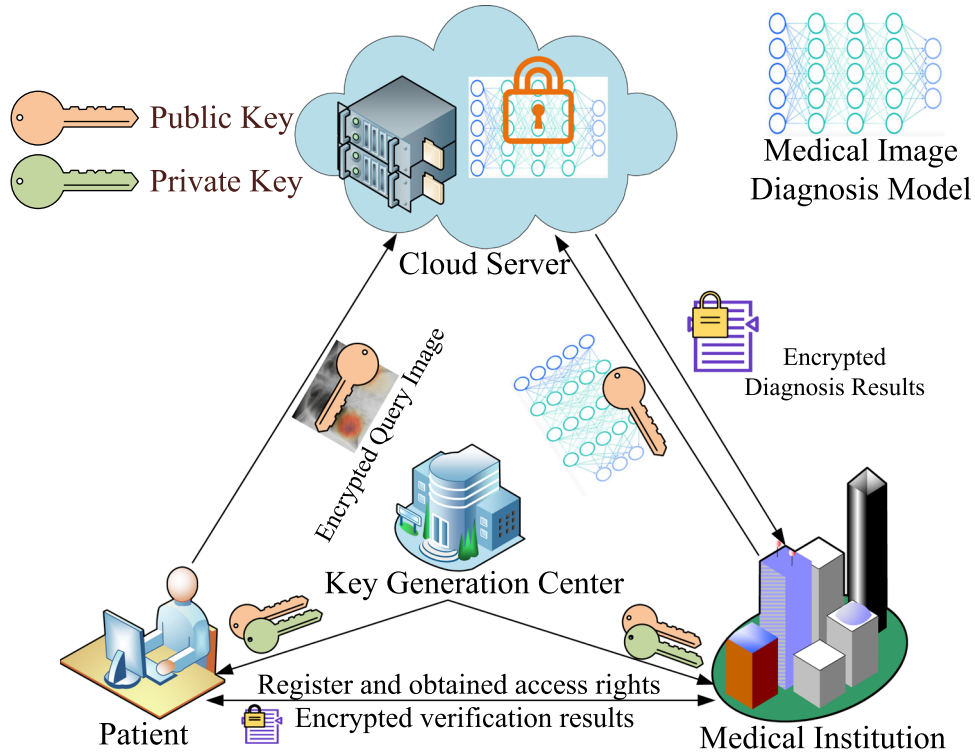
The system model is shown in Fig. 4, which includes four parties: the key generation center, the patient, the medical



(a) PneumoniaMNIST (b) PathMNIST

Fig. 3 The examples of the PneumoniaMNIST dataset and the PathMNIST dataset

Fig. 4 System model



institution, and the cloud server. The brief description of each party is as follows.

- *Key generation center*: The key generation center handles the distribution and management of all public key *pk* and private key *sk* of CKKS within the system. It is an independent and reliable entity.
- *Patient*: The patient registered with the medical institution can query their exact diseases using their provided medical image. To ensure the confidentiality of the medical image, the patient uses *pk* to encrypt their query images and sends them to the cloud server. Upon receiving the encrypted verification results from the medical institution, the patient immediately decrypts and reads them by using *sk*.
- *Medical institution*: The medical institution possesses a wealth of accurate disease diagnosis models, including medical image diagnosis models and medical record diagnosis models, and these models can offer diagnostic services to registered patients. To reduce storage overhead, the medical institution prefers to outsource these models to the cloud server. Consequently, the medical institution performs four primary functions: (1) outsourcing encrypted medical disease diagnosis models (specifically, the medical image diagnosis model is considered in this paper) to the cloud server, (2) providing patient registration systems, (3) verifying the diagnosis results, and (4) returning the encrypted verification diagnosis results to the patient. Before outsourcing the medical disease diagnosis models, the medical institution uses *pk* to encrypt and encode these models to ensure their confidentiality. Furthermore, the medical institution sends the encrypted registration information to the cloud server to validate the patient's identity. Once the medical institution receives the encrypted diagnosis results from the cloud server, it immediately decrypts the encrypted diagnosis results and then verifies the decrypted diagnosis results, and if the decrypted diagnosis results are correct, it re-encrypts the decrypted diagnosis results by using *pk* and sends them to the patient.
- *Cloud server*: The cloud server stores a large number of encrypted medical disease diagnosis models from the medical institution, and offers self-diagnosis services to reduce the diagnosis pressure on the medical institution. Consequently, the cloud server performs two main functions: (1) validating the patient's identity for their query and (2) returning the final diagnosis results based on the encrypted medical disease diagnosis models to the medical institution.

Concrete Scheme

Our work begins with the assumption that the medical image diagnosis model has been pre-trained. Our attention is concentrated on executing the diagnostic component of this existing model using homomorphically encrypted medical images. Given these premises, we introduce two innovative

CNN frameworks specifically designed to facilitate privacy-preserving diagnosis in medical imaging.

It is important to note that increasing pre-trained medical image diagnosis model depth can improve medical image diagnosis accuracy. Consequently, our goal is to substantially expand the depth of the network while reducing the computational and communication overhead for users. Nonetheless, the CKKS scheme permits only a finite set of multiplications, and deeper network architectures inherently demand an increased number of these operations. Hence, we must optimize the architecture of the pre-trained medical image model to maximize depth. Simultaneously, CKKS encryption parameters should remain minimal to maintain medical image diagnosis accuracy.

To maximize the depth of the pre-trained medical image diagnosis model, we propose three optimization technologies such as Conv-BN, Conv-AveragePool, and polynomial approximate function as activation function.

– *Batch normalization layer-convolutional layer (Conv-BN)*: BN method was proposed by Szegedy [43] in 2015 and has been widely adopted in deep learning, the purpose of which is to standardize the output of the intermediate layer of the pre-trained medical image diagnosis model, making the output of the intermediate layer more stable. Typically, the medical image of the pre-trained medical diagnosis model is standardized so that the processed sample medical images have the mean of 0 and the variance of 1. This fixed input medical image distribution aids in the stability and convergence of the algorithm. However, parameters of medical image diagnosis model are continuously updated. Even if the input medical image is standardized, the inputs to the deeper layers can still change significantly. This often results in unstable values and hampers the model's convergence. BN helps stabilize the outputs of the intermediate layers and offers the following advantages: (1) it accelerates the learning process (enabling the use of a larger learning rate), (2) it reduces the model's sensitivity to initial values, and (3) it mitigates overfitting to some extent. The format for BN is described as shown in Fig. 5, where μ_B, σ_B^2 denote the mean and the variance of mini-batch medical image m, respectively, ε is a very small value added to prevent the denominator from becoming zero, and γ and β denote the most suitable distributions for the pre-trained medical diagnosis model.

Given the output $x_i = wx + b_i$ of the convolutional layer as the input to the BN layer, the output y for each BN layer is shown in Eq. 1. This means that the medical institution can directly obtain the combined output of the convolutional layer and the BN layer for the medical image. The advantage of this method is that it ensures only one multiplication operation.

$$\begin{cases} \mu_B \leftarrow \dfrac{1}{m}\sum_{i=1}^{m} x_i \\ \sigma_B^2 \leftarrow \dfrac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2 \\ \hat{x}_i \leftarrow \dfrac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} \\ \hat{y}_i \leftarrow \gamma \hat{x}_i + \beta \end{cases}.$$

Fig. 5 Batch normalization equation

$$\begin{aligned} y_i &= \gamma_i \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} + \beta_i \\ &= \frac{\gamma_i}{\sqrt{\sigma_B^2 + \varepsilon}}(wx + b_i) + (\beta_i - \frac{\gamma_i \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}}) \\ &= \sum_i (w \times \frac{\gamma_i}{\sqrt{\sigma_B^2 + \varepsilon}})x + (\beta_i + \frac{\gamma_i \times b_i - \gamma_i \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}}). \end{aligned} \tag{1}$$

Before outsourcing the pre-trained medical image diagnosis model to the cloud server, the medical institution first encodes $w \times \frac{\gamma_i}{\sqrt{\sigma_B^2+\varepsilon}}$ and $\beta_i + \frac{\gamma_i \times b_i - \gamma_i \mu_B}{\sqrt{\sigma_B^2+\varepsilon}}$ using the CKKS algorithm. Furthermore, the medical institution encodes the mean μ_B and the variance σ_B^2 of mini-batch medical images and sends them to the cloud server for use during the encrypted medical image diagnosis stage.

– *Convolutional layer-average pooling layer (Conv-AveragePool)*: The Conv-AveragePool is also a combination block between the convolutional layer and the average pooling layer. For example, given the output $x_i = wx + b_i$ of the convolutional layer as the input to the average pooling layer, the output $y = \frac{1}{N}\sum_{i=1}^{N} x_i$ for each average

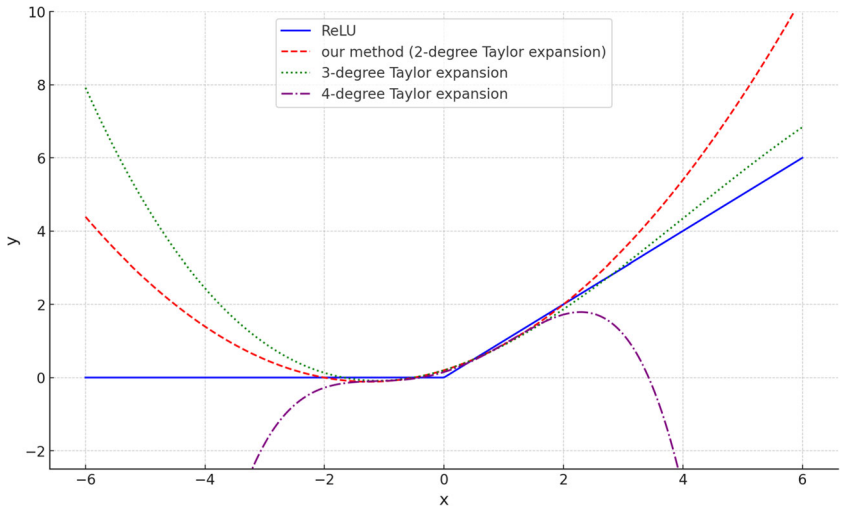


Fig. 6 The distribution between the ReLU function and polynomial approximations

pooling layer is shown in Eq. 2.

$$y = \frac{1}{N}\sum_{i=1}^{N} x_i = \frac{1}{N}\sum_{i=1}^{N}(wx + b_i) = \sum_{i=1}^{N}((\frac{1}{N}w)x + \frac{b_i}{N}). \tag{2}$$

Similar to the Conv-BN part, the medical institution uses the CKKS algorithm to encode $\frac{1}{N}w$ and $\frac{1}{N}w$, requiring only one multiplication operation between two layers of the model during the encrypted medical diagnosis stage. Furthermore, μ_B and σ_B^2 are also encoded, as they are necessary for the encrypted medical diagnosis stage.

– *Polynomial approximate function*: The activation functions are crucial in neural networks, which can help networks learn complex patterns in the data [44]. Similar to the neuron-based model of the human brain, the activation functions ultimately determine which neurons to activate in the next layer. The CNN model usually employs ReLU, Tanh, and sigmoid functions as activation functions, but these activation functions in homomorphic ciphertexts are challenging. For instance, the ReLU function is piecewise linear and involves comparisons, which cannot be performed in the ciphertext domain. To solve this issue, researchers [17–19] have proposed substituting standard activation functions with approximation polynomial methods within the context of homomorphic encryption. In our proposed scheme, the Taylor expansion of the 2-degree for the ReLU function serves as the activation function, as it provides a better approximation than 4-degree or higher-degree expansions based on experimental results, and the formula of the Taylor expansion of the 2-degree for the ReLU function is shown in Eq. 3.

$$0.1992 + 0.5002x + 0.1997x^2. \tag{3}$$

We have compared the distribution between the ReLU function, our proposed 2-degree Taylor expansion, and the other 3-degree or 4-degree Taylor expansion, as illustrated in Fig. 6, where the 3-degree Taylor expansion is Eq. 4, and the 4-degree Taylor expansion is Eq. 5. The results indicate that within the range of $[-2, 2]$, our 2-degree Taylor expansion more accurately approximates the ReLU function than the 3-degree and 4-degree Taylor expansion. Furthermore, since the pixel values of an image are confined to the range $[-1, 1]$ due to our use of BN technology, this further substantiates that our proposed 2-degree Taylor expansion can more effectively approximate the ReLU function within this range.

$$0.1995 + 0.5002x + 0.1994x^2 - 0.0164x^3. \tag{4}$$

$$0.1500 + 0.5012x + 0.2981x^2 - 0.0004x^3 - 0.0388x^4. \tag{5}$$

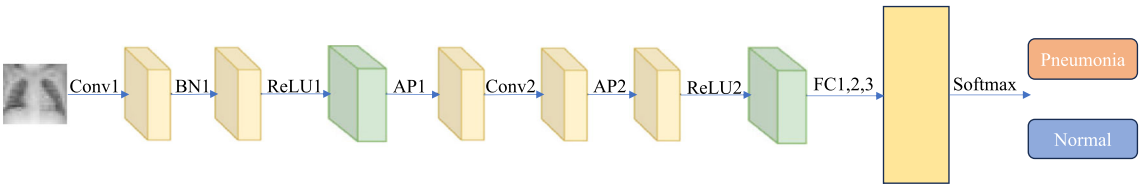


Fig. 7 PneumoniamNIST CNN architecture

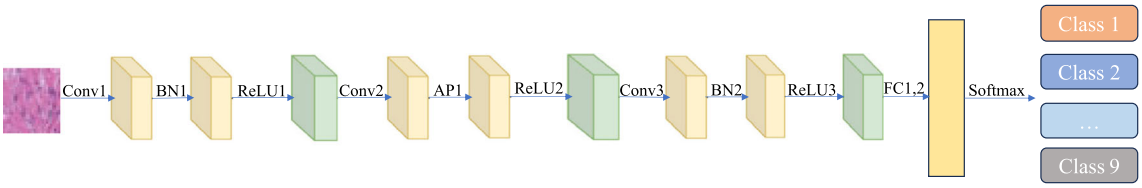


Fig. 8 PneumoniamNIST CNN architecture

Drawing on the optimization techniques previously discussed, we introduce two distinct CNN architectures tailored for the PneumoniaMNIST dataset and the PathMNIST dataset. The designs of these two specific CNN models are depicted in Figs. 7 and 8, respectively.

For the PneumoniaMNIST dataset, we build the eight-layer CNN architecture, and the detailed parameter information is shown in Table 2. In addition, we employ the 2-degree polynomial function to approximate the ReLU function, where the maximum multiplication operation is 2, and the BN layer requires 1 multiplication operation during the ciphertext medical image diagnosis stage. As shown in Fig. 7, our PneumoniaMNIST CNN architecture initially requires 12 multiplication operations (with ReLU replaced by the 2-degree polynomial function). However, by applying our proposed optimization techniques, Conv-BN and Conv-AveragePool, we can reduce the depth of multiplication operations from 12 to 10, thereby enhancing the speed of ciphertext medical diagnosis.

Likewise, we also design a CNN architecture for the PathMNIST dataset as shown in Fig. 8, which requires 13 multiplication operations, and the detailed parameter information of this architecture is shown in Table 3. To adapt the number of multiplication operations of the CKKS encryption scheme, we use our proposed optimization technologies to reduce the multiplication operations from 13 to 10.

Performance Analysis

In this section, we primarily focus on comparing the diagnostic accuracy and the diagnostic efficiency of our proposed privacy-preserving medical image diagnosis scheme with other existing schemes, such as literarures [17, 20, 32]. We use the PneumoniaMNIST dataset and the PathMNIST dataset as our experimental evaluation datasets. In addition, we employ the Pyfhel library to implement the CKKS scheme, with the following key parameters: (1) the polynomial modulus degree is set to 2^{14}, (2) the scale factor is set to 2^{40}, and (3) the maximum modulus bit length is set to 438 bits. With these parameter settings, our medical image diagnosis model is capable of supporting 11 multiplication operations in the ciphertext computation environment. Finally, the experiments are conducted on an Intel Core i7-12700 12-core CPU @2.1 GHz with 32 GB of RAM, operating under a Linux system.

Accuracy Analysis

For the Pneumonia dataset, we have built a medical image diagnosis CNN model as shown in Fig. 7. To improve model accuracy, we incorporated both convolutional layers and batch normalization (BN) with average pooling layers, thereby increasing the number of non-linear layers to

Table 2 PneumoniaMNIST CNN architecture parameter settings

	Layer	Notation	Parameters
1	Conv1	The input image size is $1 \times 32 \times 32$, the kernel size is (5, 5), the number of filters is 6, and the output size is $28 \times 28 \times 6$	Kernel and bias
2	BN1	The input size and the output size are the same values, i.e., $28 \times 28 \times 6$	Mean, variance, and random values
3	ReLU1	This layer determines which nodes can be activated by using the activation function	–
4	AP1	The kernel size is (2, 2), the stride is 2, and the output size is $14 \times 14 \times 6$	Kernel
5	Conv2	The kernel size is (5, 5), the number of filters is 16, and the output size is $10 \times 10 \times 16$	Kernel and bias
6	AP2	The kernel size is (2, 2), the stride is 2, and the output size is $5 \times 5 \times 16$	Kernel
7	ReLU2	This layer determines which nodes can be activated by using the activation function	–
8	FC1,2,3	3 fully connected layer, the first FC1 is ($16 \times 5 \times 5$, 120), the second FC2 is (120, 84), and the final FC3 is (84, 2)	Fully connection weight and bias
9	Softmax	–	–

Table 3 PathMNIST CNN architecture parameter settings

	Layer	Notation	Parameters
1	Conv1	The input image size is $3 \times 32 \times 32$, the kernel size is (4, 4), the number of filter is 64, the stride is 2 and the output size is $15 \times 15 \times 64$	Kernel and bias
2	BN1	The input size and the output size are the same values, i.e., $15 \times 15 \times 64$	Mean, variance, and random values
3	ReLU1	This layer determines which nodes can be activated by using activation function	–
4	Conv2	The kernel size is 3×3, the number of filters is 64, and the output size is $13 \times 13 \times 128$	Kernel and bias
5	BN2	The input size and the output size are the same value, i.e., $13 \times 13 \times 128$	Mean, variance, and rando values
6	ReLU2	This layer determines which nodes can be activated by using activation function	–
7	Conv3	The kernel size is 5×5, the stride is 2, the number of filters is 256, and the output size is $5 \times 5 \times 256$	Kernel and bias
8	AP1	The kernel size is 2×2, and the output size is $2 \times 2 \times 256$	Kernel
9	ReLU3	This layer determines which nodes can be activated by using activation function	–
10	FC1,2	2 fully connected layer, the first FC1 size is $(2 \times 2 \times 256, 128)$, and the final FC2 size is (128, 9)	Fully connection weight and bias
11	Softmax	–	–

two. This is a notable advancement compared to the single non-linear layer utilized in both the CryptoNet and BFVML schemes. Similarly, the MiniONN scheme also incorporated two non-linear layers; however, it leveraged secure multi-party computation technology to facilitate a deeper CNN architecture. As indicated in Table 4, our approach surpasses the performance of these comparison schemes in both plaintext and ciphertext scenarios. It is crucial to highlight that while the MiniONN and BFVML schemes achieve accuracy levels comparable to ours, the MiniONN scheme requires increased interaction between the cloud server and the patient. Moreover, the BFVML scheme is limited to supporting only one non-linear layer. These designs restrict their effectiveness on larger datasets and more complex CNN structures, potentially leading to lower accuracy.

Likewise, we also have built a medical image diagnosis CNN model for the PathMNIST dataset as shown in Fig. 8, and we have increased the number of non-linear layers to 3. The comparison of accuracy with other schemes is shown in Table 5, and we can learn our scheme achieves higher accuracy both in the plaintext and the ciphertext environments. Note that both the BFVML and CryptoNet schemes are incapable of handling multi-channel (RGB) images directly. Therefore, to accommodate these schemes, we converted the images from the PathMNIST dataset to grayscale before validating their accuracy. Although the MiniONN scheme achieved accuracy comparable to our model, it requires significantly more interactions between the cloud server and the patient. This increased interaction makes the MiniONN scheme less practical for real-world applications.

Efficiency Analysis

In this paper, our scheme uses the CKKS homomorphic encryption algorithm to encrypt the original image, in terms of encryption efficiency, model inference efficiency, and communication rounds compared with other similar works for the efficiency analysis of the scheme, which is listed in Table 6.

Table 4 Comparison of accuracy with different schemes on PneumoniaMNIST dataset

Schemes	Plaintext model	Ciphertext model
Our scheme	**85.89%**	**85.46%**
CryptoNet [17]	66.57%	65.38%
MiniONN [32]	84.81%	83.95%
BFVML [20]	83.34%	82.68%

Table 5 Comparison of accuracy with different schemes on PathMNIST dataset

Schemes	Plaintext model	Ciphertext model
Our scheme	**82.42%**	**82.15%**
CryptoNet [17]	60.25%	59.64%
MiniONN [32]	80.57%	80.18%
BFVML [20]	60.69%	60.25%

Table 6 Comparison efficiency with other schemes. n is the depth of the CNN model

Schemes	Datasets	Encryption time	Inference time	Communication rounds
Our scheme	PneumoniaMNIST	**11 *s***	**128 *s***	**1**
	PathMNIST	58 s	386 s	1
CryptoNet [17]	PneumoniaMNIST	23 s	252 s	1
BFVML [20]	PneumoniaMNIST	15 s	192 s	1
MiniONN [32]	PneumoniaMNIST	302 s	468 s	n
	PathMNIST	827 s	1241 s	n

For encryption efficiency, we use Pyfhel[1] library to achieve the CKKS scheme, the BFV scheme, and the BGV scheme while using phe[2] library to achieve Paillier's scheme. From Table 6, we can learn that the encryption of one image in the PneumoniaMNIST dataset and the PathMNIST dataset is 11 s and 58 s, which is lower than that of the comparative schemes.

For model inference efficiency, the inference time of our scheme for one image in the PneumoniaMNIST dataset and the PathMNIST dataset is 128 s and 326 s, respectively, which is much faster than that of the comparative schemes due to our proposed layer optimization technology.

For communication rounds, our scheme, CryptoNet scheme, and BFVML scheme are lower than the MiniONN scheme, because the MiniONN scheme uses secure multi-party protocols to achieve non-linear layer computation.

Conclusion

In this paper, we proposed a robust privacy-preserving medical image diagnosis scheme that not only ensures the confidentiality of the patient's medical images and the parameters of the medical image diagnosis model, but also achieves higher accuracy and speeds up the diagnostic process. Specifically, we utilized the CKKS scheme to encrypt both the patient's medical image and the medical image diagnosis model, safeguarding the confidentiality of both the patient's medical images and the medical image diagnosis model at the same time. In addition, we incorporated batch normalization to support the use of multiple non-linear layers in deep convolutional neural networks within a ciphertext environment. To handle non-linear functions in the ciphertext field, we employed a 2-degree polynomial function to effectively approximate the ReLU activation function. To address the issue of insufficient CNN depth, we introduced two innovative network depth optimization techniques: Conv-BN and Conv-AveragePool. Overall, our proposed scheme demonstrates robustness and efficiency, as substantiated through extensive experimental analysis.

[1] Available at https://github.com/ibarrond/Pyfhel.

[2] Available at https://github.com/data61.

Author Contribution Jiuhong Ran and Dong Li agreed on the content of the study. Jiuhong Ran collected all the data for analysis and designed this method, and Dong Li completed the experimental analysis based on agreed steps. Results and conclusions are discussed and written together. The authors read and approved the final manuscript.

Funding This work was supported in part by the Natural Science Foundation of Chongqing (Innovation and Development Joint Fund) under Grant CSTB2023NSCQ-LZX0149, in part by the China Scholarship Council, and in part by the National Natural Science Foundation of China under Grant U20A20176.

Data Availability The PneumoniaMNIST dataset and the PathMNIST dataset are available at https://github.com/MedMNIST/MedMNIST.

Declarations

Ethics Approval and Consent to Participate Not applicable

Conflict of Interest The authors declare no competing interests.

Human and Animal Rights This article does not contain any studies with human or animal subjects performed by any of the authors.

References

1. Sharifi, Abbas and Ahmadi, Mohsen and Mehni, Mohammad Amin and Jafarzadeh Ghoushchi, Saeid and et al., "Experimental and numerical diagnosis of fatigue foot using convolutional neural network," Computer Methods in Biomechanics and Biomedical Engineering, 24, 844-848, Taylor & Francis (2021).
2. Veluchamy, S and Sudharson, S and Annamalai, R and Bassfar, Zaid and Aljaedi, Amer and et al., "Automated detection of COVID-19 from multimodal imaging data using optimized convolutional neural network model," Journal of Imaging Informatics in Medicine, 1-15 (2024).
3. Yadav, Samir S and Jadhav, Shivajirao M, "Deep convolutional neural network based medical image classification for disease diagnosis," Journal of Big Data, 6, 1-18 (2019).
4. Gómez-Flores, Wilfrido and Pereira, Wagner Coelho de Albuquerque, "Gray-to-color image conversion in the classification of breast lesions on ultrasound using pre-trained deep neural networks," Medical & Biological Engineering & Computing, 61, 3193-3207 (2023).

5. Chen, LiFang and Li, Jiawei and Ge, Hongze, "TBUnet: A pure convolutional U-Net capable of multifaceted feature extraction for medical image segmentation," Journal of Medical Systems, 47, 122 (2023).
6. Rajeev, R and Samath, J Abdul and Karthikeyan, NK, "An intelligent recurrent neural network with long short-term memory (LSTM) BASED batch normalization for medical image denoising," Journal of Medical Systems, 43, 234 (2019).
7. Kermouni Serradj, Nadia and Messadi, Mahammed and Lazzouni, Sihem, "Classification of mammographic ROI for microcalcification detection using multifractal approach, Journal of Digital Imaging, 35, 1544-1559 (2022).
8. Moradi, Sasan and Brandner, Christoph and Spielvogel, Clemens and Krajnc, Denis and et al., "Clinical data classification with noisy intermediate scale quantum computers," Scientific Reports, 12, 1851 (2022).
9. Shang, Yong and Gao, Xing and An, Aimin, "Multi-band spatial feature extraction and classification for motor imaging EEG signals based on OSFBCSP-GAO-SVM model: EEG signal processing," Medical & Biological Engineering & Computing, 61, 1581-1602 (2023).
10. Radović, Nevena and Prelević, Vladimir and Erceg, Milena and Antunović, Tanja, "Machine learning approach in mortality rate prediction for hemodialysis patients," Computer Methods in Biomechanics and Biomedical Engineering, 25, 111-122, Taylor & Francis (2022).
11. Baji, Faiq Sabbar and Abdullah, Saleema Baji and Abdulsattar, Fatimah S, "K-mean clustering and local binary pattern techniques for automatic brain tumor detection," Bulletin of Electrical Engineering and Informatics, 12, 1586-1594 (2023).
12. DL, M and DP, M, "Processing of clinical notes for efficient diagnosis with feedback attention–based BiLSTM," Journal of Digital Imaging, 36,1431 (2023).
13. Stahl, Bernd Carsten and Wright, David, "Ethics and privacy in AI and big data: Implementing responsible research and innovation," IEEE Security & Privacy, 16, 26-33 (2018).
14. Tanuwidjaja, Harry Chandra and Choi, Rakyong and Baek, Seunggeun and Kim, Kwangjo, "Privacy-preserving deep learning on machine learning as a service: a comprehensive survey," IEEE Access, 8, 167425-167447 (2020).
15. Pulido-Gaytan, Bernardo and Tchernykh, Andrei and Cortés-Mendoza, Jorge M and Babenko, Mikhail and Radchenko, Gleb and et al., "Privacy-preserving neural networks with homomorphic encryption: Challenges and opportunities," Peer-to-Peer Networking and Applications, 14, 1666-1691 (2021).
16. de Almeida, Patricia Gomes Rêgo and dos Santos, Carlos Denner and Farias, Josivania Silva, "Artificial intelligence regulation: a framework for governance," Ethics and Information Technology, 23, 505-525 (2021).
17. Gilad-Bachrach, Ran and Dowlin, Nathan and Laine, Kim and Lauter, Kristin and Naehrig, Michael and Wernsing, John, "Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy," in Proceedings of the International Conference on Machine Learning, 201-210, PMLR (2016),
18. Chou, Edward and Beal, Josh and Levy, Daniel and Yeung, Serena and Haque, Albert and Fei-Fei, Li, "Faster cryptonets: Leveraging sparsity for real-world encrypted inference," in arXiv:1811.09953
19. Hesamifard, Ehsan and Takabi, Hassan and Ghasemi, Mehdi, "Cryptodl: Deep neural networks over encrypted data," in arXiv:1711.05189
20. Falcetta, Alessandro and Roveri, Manuel, "Privacy-preserving deep learning with homomorphic encryption: An introduction," IEEE Computational Intelligence Magazine, 17, 14-25 (2022)
21. Cheon, Jung Hee and Kim, Dongwoo and Kim, Duhyeong and Lee, Hun Hee and Lee, Keewoo, "Numerical method for comparison on homomorphically encrypted numbers," in Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security, 415-445, Springer (2019).
22. Rivest, Ronald L and Adleman, Len and Dertouzos, Michael L and others, "On data banks and privacy homomorphisms," Foundations of Secure Computation, 4, 169-180 (1978).
23. Naccache, David and Stern, Jacques, "A new public key cryptosystem based on higher residues," in Proceedings of the 5th ACM Conference on Computer and Communications Security, 59-66, ACM (1998).
24. Okamoto, Tatsuaki and Uchiyama, Shigenori "A new public-key cryptosystem as secure as factoring," in Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques, 308-318, Springer (1998).
25. Paillier, Pascal, "Public-key cryptosystems based on composite degree residuosity classes," in Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, 223-238, Springer (1999).
26. ElGamal, Taher, "A public key cryptosystem and a signature scheme based on discrete logarithms," IEEE Transactions on Information Theory, 31, 469-472 (1985).
27. Gentry, Craig, "Fully homomorphic encryption using ideal lattices," in Proceedings of the 41th Annual ACM Symposium on Theory of Computing, 169-178, ACM (2009).
28. Van Dijk, Marten and Gentry, Craig and Halevi, Shai and Vaikuntanathan, Vinod, "Fully homomorphic encryption over the integers," in Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, 24-43, Springer (2010).
29. Brakerski, Zvika and Gentry, Craig and Vaikuntanathan, Vinod, "(Leveled) fully homomorphic encryption without bootstrapping," ACM Transactions on Computation Theory, 6, 1-36 (2014).
30. Bos, Joppe W, Lauter, Kristin, Loftus, Jake and Naehrig, Michael, "Improved security for a ring-based fully homomorphic encryption scheme," Cryptography and Coding, 45-64 (2013)
31. Fan, Junfeng and Vercauteren, Frederik, "Somewhat practical fully homomorphic encryption," Cryptology ePrint Archive, 2012.
32. Liu, Jian and Juuti, Mika and Lu, Yao and Asokan, Nadarajah, "Oblivious neural network predictions via minionn transformations," in Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 619-631, ACM (2017).
33. Bourse, Florian and Minelli, Michele and Minihold, Matthias and Paillier, Pascal, "Fast homomorphic evaluation of deep discretized neural networks," in Proceedings of the 2018 38th Annual International Cryptology Conference, 483-512, Springer (2018).
34. Wang, Yu and Chen, Liquan and Wu, Ge and Yu, Kunliang and Lu, Tianyu, "Efficient and secure content-based image retrieval with deep neural networks in the mobile cloud computing," Computers & Security, 128, 103163 (2023).
35. Boemer, Fabian and Costache, Anamaria and Cammarota, Rosario and Wierzynski, Casimir, "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, 45-56, ACM (2019).
36. Mishra, Pratyush and Lehmkuhl, Ryan and Srinivasan, Akshayaram and Zheng, Wenting and Popa, Raluca Ada, "Delphi: A cryptographic inference system for neural networks," in Proceedings of the 2020 Workshop on Privacy-Preserving Machine Learning in Practice, 27-30, ACM (2020).
37. Yao, Andrew C, "Protocols for secure computations," in Proceedings of the 1982 23rd Annual Symposium on Foundations of Computer Science, 160-164, IEEE (1982).
38. Barni, Mauro and Orlandi, Claudio and Piva, Alessandro, "A privacy-preserving protocol for neural-network-based computation," in Proceedings of the 8th Workshop on Multimedia and Security, 146-151, ACM (2006).

39. Mohassel, Payman and Zhang, Yupeng, “Secureml: A system for scalable privacy-preserving machine learning," in Proceedings of the 2017 IEEE Symposium on Security and Privacy, 19-38, IEEE (2017).
40. Rouhani, Bita Darvish and Riazi, M Sadegh and Koushanfar, Farinaz, “Deepsecure: Scalable provably-secure deep learning," in Proceedings of the 55th Annual Design Automation Conference, 1-6, ACM (2018).
41. Juvekar, Chiraag and Vaikuntanathan, Vinod and Chandrakasan, Anantha, “{GAZELLE}: A low latency framework for secure neural network inference," in Proceedings of the 2018 27th USENIX Security Symposium, 1651-1669, USENIX (2018).
42. Yang, Jiancheng and Shi, Rui and Wei, Donglai and Liu, Zequan and Zhao, Lin and et al., “Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification," Scientific Data, 10, 41 (2023).
43. Ioffe Sergey and Christian Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift," in Proceedings of the International Conference on Machine Learning, 448-456, PMLR (2015).
44. Hassan, Abdelrhman and Liu, Fei and Wang, Fanchuan and Wang, Yong, “Secure content based image retrieval for mobile users with deep neural networks in the cloud," Journal of Systems Architecture, 116, 102043 (2021).
45. Ibarrondo, Alberto and Viand, Alexander, “Pyfhel: Python for homomorphic encryption libraries," in Proceedings of the 9th on Workshop on Encrypted Computing & Applied Homomorphic Cryptography, 11-16, ACM (2021).