Computers and Artificial Intelligence；Vol.3, No.3; 2026
E-ISSN: 3006 -3728 P-ISSN：3006-371X
DOI: https://doi.org/10.70267/cai.26v3n3.155166
Published by: Zeus Press

An Optimization Study of TFHE Encryption Algorithm Based on tiny-DNN

Penghui Liu*

School of Information, Beijing Forestry University, BeiJing, China

**Corresponding author: Penghui Liu.*

Abstract

Privacy leakage, malicious tampering, and incompleteness of big data have become severe challenges for deep learning. Meanwhile, fully homomorphic encryption algorithms suffer from increased energy consumption and insufficient precision. To address these issues, this paper proposes a dual-path heterogeneous privacy-preserving inference mechanism. First, data is divided into private and non-private categories using a privacy segmentation coefficient. Private data is placed in the TFHE encryption domain, while a tiny-DNN-based secure black-box is employed to protect non-private data. The zero-error nonlinear activation mechanism of the MUX logic gate in TFHE is used to train the deep learning framework. Subsequently, a cross-domain fixed-point homomorphic adder fuses the two types of data to obtain the ciphertext result formed by end-to-end heterogeneous data. This approach ensures the prediction accuracy of deep learning while reducing the network latency of big data classification. Experimental results show that the proposed system reduces end-to-end inference latency to 970 ms while maintaining model accuracy. Security analysis demonstrates that the system possesses strong robustness against attacks, thereby providing confidentiality and integrity protection for real-time inference of big data.

Keywords

deep learning framework, fully homomorphic encryption, data security, MUX logic gate

1. Introduction

In the information age dominated by big data, privacy and security are of paramount importance [1], especially given the risks of leakage and tampering of private data [2]. When focusing on the static storage and transmission of data, traditional encryption algorithms such as AES and 3DES can effectively address related issues. However, ciphertext in the decryption process is vulnerable to attacks and threats, introducing significant security risks. Rivest, Adleman, and Dertouzos first introduced the concept of homomorphic encryption (HE) [3,4], with the computational formula: $g(f(x) \oplus f(y)) = g(f(x)) \oplus g(f(y))$. Here, f represents the encryption function, which converts plaintext $x$and $y$into ciphertext $f(x)$and $f(y)g$ represents the decryption function, which restores the ciphertext to plaintext. Although homomorphic encryption solves the problem of direct computation on plaintext, it still has clear limitations: it can only support a single type of

operation, cannot handle complex hybrid operations, and incurs increased computational overhead, making it difficult to meet the demands of multi-step and multi-operation computations on ciphertext in real-world scenarios. To overcome these limitations, fully homomorphic encryption (FHE) [5], as an optimized scheme of homomorphic encryption, supports arbitrary numbers of additions, multiplications, and hybrid operations on the basis of direct computation on ciphertext (where the decrypted result is consistent with the plaintext computation). It truly realizes the computational paradigm of performing arbitrarily complex processing on ciphertext, making FHE a key technology for solving privacy protection problems. Building on this foundation, this paper improves the fully homomorphic encryption algorithm and proposes a TFHE scheme applied to private data. For the non-private portion, the deep learning framework is modified and enhanced by introducing a tiny-DNN-based secure black-box algorithm. Finally, the MUX logic gate is used to complete the fusion of the two types of data. The main contributions of this paper are as follows:

- This paper proposes a dual-path hierarchical preprocessing mechanism based on private data, placing private data in the TFHE encryption domain and non-private data in the plaintext domain to address the high latency issue of TFHE.
- This paper proposes a cross-domain feature-level fusion mechanism to achieve heterogeneous fusion between the encryption domain and the plaintext domain.
- Based on the logic gate truncation mechanism of the multiplexer (MUX), this paper implements zero-error computation of nonlinear activation functions on data within the encryption domain.
- For the protection of non-private data, this paper designs a secure black-box on top of the deep learning framework, further realizing the protection of non-private data in the plaintext domain.

2. Related Technologies

2.1 AutoFHE: Automated Adaption of CNNs for Efficient Evaluation over FHE [6]

This work adaptively adjusts Convolutional Neural Networks (CNNs) [7] into an efficient framework for fully homomorphic encryption (FHE) environments. Its core objective is to balance efficiency and accuracy during secure inference under the RNS-CKKS encryption scheme. The method employs hierarchical mixed-degree polynomials to replace traditional nonlinear activation functions and automatically optimizes the polynomial degree and coefficients according to the varying sensitivity of each layer to approximation errors. In addition, AutoFHE jointly optimizes polynomial search and homomorphic evaluation architecture, using a multi-objective optimization algorithm to maximize accuracy while minimizing the number of bootstrapping operations.

However, the CKKS scheme adopted in this paper is an approximate homomorphic encryption scheme. Its underlying mathematical principles only support addition and multiplication over real or complex numbers and cannot directly compute nonlinear activation functions with inflection points (such as the ReLU function). Consequently, such methods are forced to use high-degree polynomials to approximate activation functions. Yet, polynomial fitting is inherently a smooth curve, which inevitably introduces irreducible errors at the inflection point ($x = 0$). Furthermore, as the number of layers in deep learning networks increases, these errors gradually accumulate and amplify, ultimately leading to a decline in model inference accuracy.

2.2 REDsec: Running Encrypted Discretized Neural Networks in Seconds [8]

This paper aims to address the challenge of balancing security and efficiency in privacy-preserving inference of deep neural networks and proposes an implementation based on FHE. First, from the perspective of data privacy protection, the FHE algorithm is used to fully homomorphically encrypt all inference data and model parameters. Second, to mitigate the performance overhead caused by encryption, auxiliary techniques such as arithmetic circuit optimization are introduced. Then, a programmable bootstrapping mechanism is incorporated to suppress noise accumulation during encryption. Finally, an adapted inference pipeline is designed to reduce redundant encryption and decryption operations.

It should be noted that this work encrypts all input data into the TFHE domain. Although individual logic gate operations in FHE are relatively fast, the scheme requires evaluating millions to billions of logic gates,

resulting in high overall network latency. Moreover, when dealing with large-size images, the full-image encryption approach lacks scalability.

2.3 Fast and Private Inference of Deep Neural Networks by Co-designing Activation Functions [9]

This paper optimizes the computational bottleneck of nonlinear activation functions in fully homomorphic encryption inference through the co-design of algorithms and encryption parameters. To reduce the massive computational overhead in FHE, the work abandons the standard ReLU activation function [10] and instead designs and searches for specific low-degree polynomials during the model training phase for replacement and approximation. The core idea is to find a more optimized static trade-off between the execution efficiency of ciphertext computation and the prediction accuracy of the deep learning model, thereby alleviating the latency pressure of ciphertext inference.

Although the paper makes progress in optimizing activation functions within the pure ciphertext domain, it still fails to break free from the limitations of traditional fully homomorphic encryption paradigms. It has two core pain points: First, the inherent and irreversible precision loss caused by polynomial approximation. No matter how the co-design optimizes the polynomial degree and coefficients, a continuous polynomial curve cannot mathematically perfectly realize the nonlinear truncation of the ReLU function at zero. This approximation error accumulates severely layer by layer in deep networks, directly causing a decline in the final prediction accuracy of the model. Second, there is global redundancy in computational resource allocation. Although the method optimizes the computational cost of a single activation function, it still adopts the coarse-grained strategy of full data encryption. The system forcibly places non-private background features which constitute the dominant proportion of an image into the ciphertext domain for polynomial evaluation. This not only wastes valuable homomorphic computing resources but also makes it difficult for the overall end-to-end system inference latency to meet practical deployment requirements.

2.4 Research Motivation

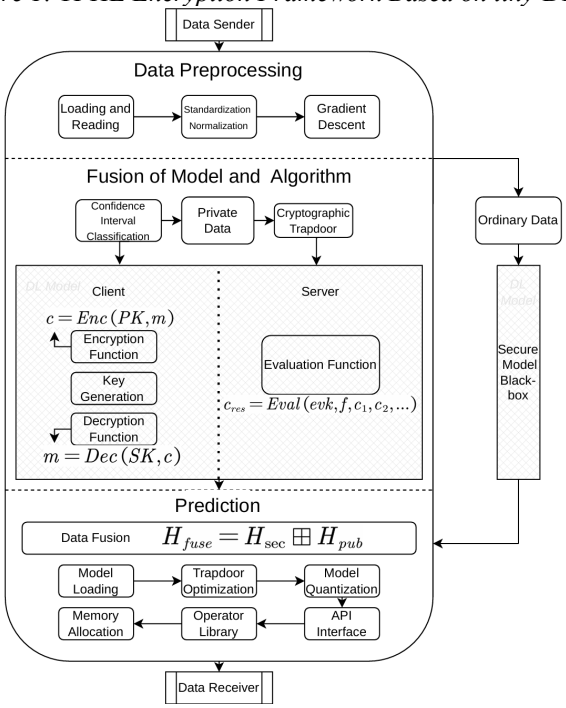
In current research on privacy-preserving machine learning and homomorphic encryption inference, although frontier works such as AutoFHE and REDsec have made significant progress in accelerating nonlinear layer computations, they still expose several limitations that restrict the accuracy and efficiency of homomorphic encryption inference in practical applications. First, the CKKS scheme used by AutoFHE only supports addition and multiplication of decimal data. Existing studies employ polynomial approximation methods (e.g., EvoReLU) [11] to fit the ReLU function. Since the image of a polynomial approximation is a smooth curve, it can never perfectly coincide with the ReLU function $f(x) = \max(0, x)$, inevitably introducing fitting errors. Second, REDsec encrypts all input data to maximize privacy protection. In large-scale scenarios, this leads to exponential growth in encryption computation, causing significant latency challenges—specifically, inference times can reach the hour level.

Based on the above problems, this paper proposes a reconstructed ciphertext inference framework for privacy-preserving deep learning. The method achieves encryption and decryption computations within polynomial time from two dimensions—database and cryptographic operators—thereby solving the issues of high computational complexity and large energy consumption in traditional homomorphic encryption algorithms. Through a privacy segmentation coefficient, big data is divided into private data and ordinary (non-private) data, with the two parts adopting different encryption mechanisms. Private data uses an optimized TFHE scheme, while ordinary data employs a tiny-DNN-based data encryption mechanism—i.e., a secure black-box model designed on top of the deep learning framework. This approach reduces computational and communication overhead while avoiding the problems of polynomial overfitting and inaccuracy, providing important research value and significance for security in the era of big data.

3. System Framework

This chapter proposes a TFHE encryption framework based on tiny-DNN, as shown in Figure 1.

Figure 1: TFHE Encryption Framework Based on tiny-DNN



As illustrated in Figure 1, the proposed system framework consists of three core modules: data preprocessing and classification, TFHE-DNN algorithm fusion, and model prediction. In the data preprocessing module, during data loading and reading, automatic batching and random shuffling are performed. To eliminate dimensional differences in the data, standardization and normalization operations are applied using standard deviation and 0-1 normalization. The data is also divided into mini-batches to reduce memory usage during each training iteration. Additionally, the data is classified into private and non-private categories based on its privacy characteristics. The model and algorithm fusion module adopts a dual-path branch design. Private data is encrypted using TFHE on the user side and undergoes homomorphic evaluation on the server side. Non-private data is fed into a secure model black-box. Feature extraction in both branches is based on multi-channel convolutional operations, where multiple sets of convolution kernels are used to extract multi-dimensional features from the input and generate corresponding feature maps. The prediction module completes dual-path feature fusion and model optimization deployment, ultimately outputting the final inference result. Specifically, in the data preprocessing stage, the data is divided into two categories according to its privacy attributes: the first category is private data, which is encrypted using the TFHE scheme; the second category is non-private data, which does not need to enter the FHE encryption domain. For private data, this paper abandons the traditional polynomial approximation approach. Leveraging the advantage of Boolean logic supported by FHE, a logic gate truncation mechanism based on multiplexers (MUX) is designed to achieve zero-error nonlinear activation function computation in the encryption domain. For non-private data, although FHE encryption is not applied, a deep learning-based secure model black-box is designed to ensure a certain level of security. Finally, this paper proposes a cross-domain data fusion scheme between the encryption domain and the plaintext domain. By utilizing the ciphertext-plaintext homomorphic operation capability supported by the FHE scheme, the public features extracted from the plaintext domain are encoded as constant terms in the encryption domain and then homomorphically fused with the user privacy features extracted from the encryption domain.

4. Research on Data Encryption Mechanism Based on tiny-DNN

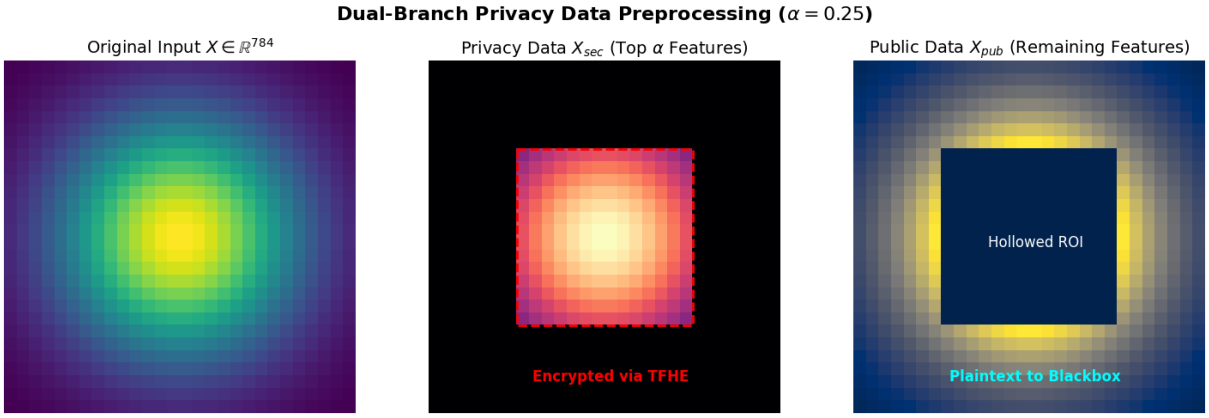
4.1 Dual-path Hierarchical Preprocessing Mechanism Based on Private Data

For the private portions of images, this paper employs the TFHE algorithm for data encryption, while ordinary (non-private) data is processed using a secure black-box mechanism. Regarding image segmentation,

directly traversing the entire plaintext image would lead to the leakage of private data. Therefore, this paper performs preprocessing directly on the TFHE client side.

Assume the privacy segmentation coefficient (Privacy Split Ratio, i.e., the highest information entropy [12] data in the input image is placed in the encryption domain) $\alpha = 0.25$. The resulting segmented image is shown in Figure 2.

Figure 2: Classification Processing Module



The execution process of the dual-path hierarchical preprocessing mechanism is as follows:

1. Data is input by the user to the local client. The system first performs lightweight feature importance scoring on the original input vector $X \in \mathbb{R}^N$. Considering the huge computational overhead that would be introduced by deploying complex semantic segmentation models locally, this paper abandons deep learning-based segmentation and instead adopts statistical feature evaluation based on prior masks. Taking the MNIST dataset used in this paper as an example, the core information of handwritten digits is typically concentrated in the center of the image. The system calculates the importance weight vector W_{imp} for each dimension accordingly.

2. The expressions for the private data subset X_{sec} and the public data subset X_{pub} are shown in Equation (1):

$$\begin{cases} X_{\text{sec}} = \{x_i \in X \mid W_{\text{imp}}(x_i) \text{ is in the top } \alpha \text{ proportion}\} \\ X_{\text{pub}} = X \setminus X_{\text{sec}} \end{cases} \tag{1}$$

where W_{imp} is the importance weight vector for each dimension.

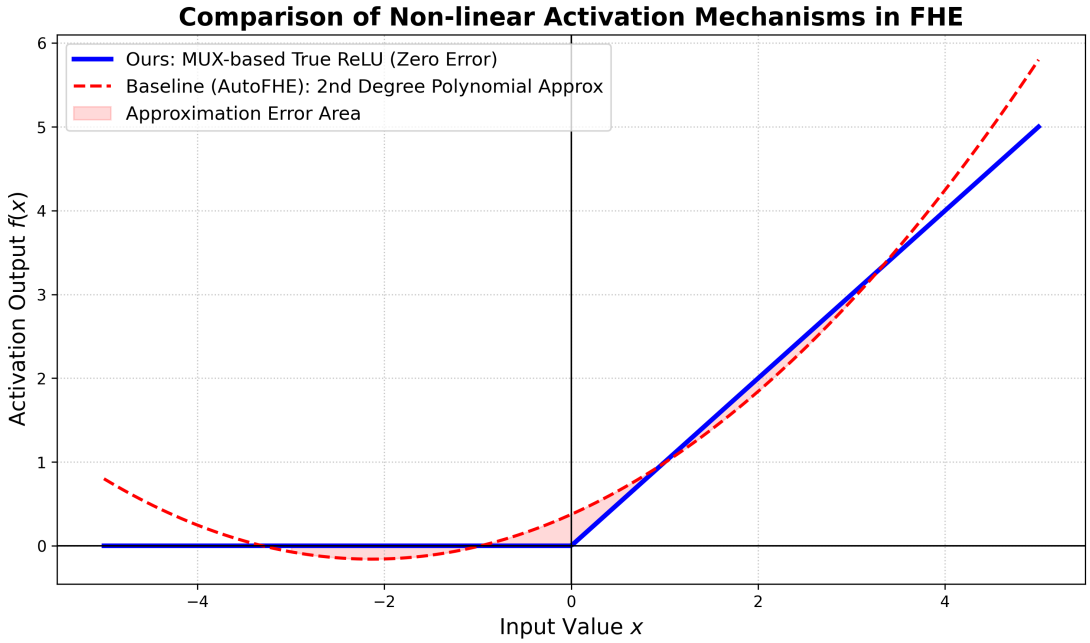
3. After completing the segmentation of the input data on the client side, the system uses the TFHE public key pk to encrypt the high-privacy data X_{sec}, generating the ciphertext tensor $C_{\text{sec}} = Enc_{pk}(X_{\text{sec}})$. For X_{pub}, the system retains its plaintext state. Finally, the system transmits the heterogeneous data pair $\{C_{\text{sec}}, X_{\text{pub}}\}$ to the server over the physical network.

4.2 MUX-based Logic Gate Truncation Mechanism

This paper leverages the advantage of Boolean circuit operations supported by the TFHE scheme and designs a nonlinear truncation mechanism for encrypted data based on multiplexers. This mechanism approximates the ReLU function according to the sign bit of the data, achieving zero-error nonlinear transformation in the encryption domain.

Figure 3 compares the activation effects of traditional polynomial fitting and the proposed MUX truncation mechanism. Polynomial fitting produces an error region near $x = 0$ (red area), whereas the proposed mechanism perfectly coincides with the true ReLU polyline, eliminating approximation errors.

Figure 3: Comparison of Activation Function Results



The execution process of the MUX truncation mechanism is as follows:

1. This paper converts signed integers into binary form and maps them in the ciphertext domain. The input data is transformed into two's complement form. In the ciphertext domain, a scalar datum is mapped to a sequence containing 8 independent ciphertext blocks, i.e., $C = \{c_0, c_1, \ldots, c_7\}$, where $c_7$is the most significant bit.
2. When data $x \geq 0$, its sign bit is 0; when $x < 0$, the sign bit is 1. Thus, the cloud server only needs to extract the highest bit $c_7$of the ciphertext sequence to use the data as a control signal for circuit on/off without seeing the original information.
3. The MUX logic gate operator provided by TFHE is introduced. The system extracts the sign bit ciphertext $c_7$and connects it to the select control port of the MUX gate, while the constant zero ciphertext $Enc_{pk}(0)$and the original input ciphertext $C_{in}$are connected to the data path ports. The MUX gate performs the following computation for each bit in the ciphertext state:

$$C_{out}^{(i)} = \text{MUX}\left(c_7, Enc_{pk}(0), c_{in}^{(i)}\right), i \in \{0,1,\ldots,7\} \quad (2)$$

The meaning of this formula is: when the control bit c_7= 1, the MUX circuit connects to the $Enc_{pk}(0)$port and directly outputs 0; when c_7= 0, the circuit maintains the original path and outputs C_{in}as is. Based on the above circuit logic, this paper reduces the nonlinear activation computation error to 0 and simplifies the computationally expensive activation layer into a streamlined operation that requires only 8 MUX logic gate executions, thereby improving the operational efficiency of ciphertext inference.

4.3 Secure Black-box Optimization Mechanism Based on Deep Learning Framework

The dual-path hierarchical preprocessing mechanism proposed in this paper removes the large proportion of non-private data $X_{\text{pub}}$from the input, accelerating the inference speed. To ensure the overall security of the system, this paper designs a secure black-box based on a deep learning model. This mechanism aims to restrict the cloud server's access to and tampering with X_{pub}data. The execution process of the secure black-box is as follows:

1. Since X_{pub}exists in plaintext in the non-encrypted domain, the black-box performs a simple integrity check immediately upon receiving the data. The system computes the cumulative checksum of the input data to verify whether the data has been eavesdropped on or tampered with during transmission:

$$v_{\text{check}} = \text{Hash}(X_{\text{pub}}) \tag{3}$$

Only when the check passes can the data enter the secure black-box. This step constructs a defensive barrier at the entrance.

2. The verified non-private data is then fed into the plaintext neural network branch f_{pub} inside the black-box. The neural network completes the forward propagation. The network weights W_{pub} are retained only within the black-box, while the cloud server is only granted permission to trigger the computation and has no way to reverse-engineer the data inside the network.
3. The neural network inside the black-box must also keep the number of neurons in the output layer, the number of categories in the current deep learning classification task, and the number of outputs in the ciphertext FHE consistent. This allows the system to compress a large portion of non-private data into low-dimensional classification feature data. The output feature vector of the plaintext black-box H_{pub}can be expressed as:

$$H_{\text{pub}} = f_{\text{pub}}(X_{\text{pub}}; W_{\text{pub}}), H_{\text{pub}} \in \mathbb{R}^K \tag{4}$$

where $K$is the total number of categories in the target task.

4.4 Cross-domain Feature-level Heterogeneous Data Fusion Mechanism

The dual-path hierarchical feature preprocessing mechanism divides the original input data into two parts. It is ultimately necessary to securely and losslessly fuse the private and non-private data. This paper designs a cross-domain feature-level heterogeneous fusion mechanism that utilizes a ciphertext ripple-carry adder [13] to achieve the fusion of the two types of data without exposing the data. The execution process of this fusion mechanism is as follows:

1. The non-private data is directly encoded and converted into ciphertext using the evaluation key. After the secure black-box outputs H_{pub}, the cloud server converts it into ciphertext form according to bit size, making H_{pub}consistent in format with H_{sec}.
2. After unifying the data format, a homomorphic full adder circuit is constructed using the basic logic gates (AND,OR,XOR) in TFHE. Let the i-th bits of $H_{\text{sec}}$and H_{pub}be $a_i$and $b_i$respectively, and the current carry ciphertext be c_i. The mathematical logic of data fusion can be expressed as:

$$\begin{cases} S_i = a_i \oplus b_i \oplus c_i \\ c_{i+1} = (a_i \otimes b_i) \vee (c_i \otimes (a_i \oplus b_i)) \end{cases} \tag{5}$$

where $\oplus$denotes bootsXOR, $\otimes$denotes bootsAND, and $\vee$denotes bootsOR. By executing the above formula bit by bit, the cloud can compute the fused ciphertext H_{fuse}without accessing any plaintext data, i.e., $H_{\text{fuse}} = H_{\text{sec}} \boxplus H_{\text{pub}}$($\boxplus$ denotes the homomorphic addition operator). Finally, the cloud returns the fused ciphertext prediction result $H_{\text{fuse}}$to the user. The user decrypts it locally using the private key to obtain the end-to-end deep learning prediction result.

5. Simulation Experiments and Performance Evaluation

5.1 Parameter Settings

The parameter settings for the simulation experiments are shown in Table 1.

Table 1: Software and Hardware Experimental Environment Configuration

Params.	Values	Params.	Values
Operating System	Windows 11 (64-bit)	Terminal Environment	Windows PowerShell
Compilation Toolchain	MSYS2 (UCRT64)	Compiler	GCC (g++ 15.2.0)
Build Tool	CMake	Language Standard	C++14
Development Environment	Visual Studio Code	Homomorphic Encryption Library	TFHE
Inference Framework	tiny-dnn	Hardware Model	Legion Y7000P IRX9
CPU	Intel Core i7-14700HX	GPU	RTX 4070 Laptop

5.2 Evaluation Metrics

This paper comprehensively compares state-of-the-art schemes from related technologies and adopts model accuracy and end-to-end inference latency as the core evaluation metrics.

1. Model Accuracy Existing schemes typically use polynomial approximation for activation functions, which inevitably introduces approximation errors that degrade accuracy. This paper uses ciphertext accuracy to evaluate the usability of the scheme. The calculation formula is as follows:

$$Accuracy = \frac{N_{correct}}{N_{total}} \times 100\% \tag{6}$$

where N_{total}represents the total number of samples in the test set, and $N_{correct}$represents the number of samples that are completely correctly classified by the model after data classification, cloud computation, and other processes. This metric directly reflects the system's ability to extract features from the original data.

2. End-to-End Inference Latency Due to the high overhead of polynomial multiplication and bootstrapping operations in FHE, this metric effectively evaluates whether a privacy-preserving system has practical value. End-to-end latency is expressed as the sum of encryption, cloud computation, and decryption times. For the dual-path heterogeneous fusion architecture proposed in this paper, the cloud inference time $T_{cloud}$is further reduced. Since the plaintext TEE black-box branch and the TFHE ciphertext branch execute in parallel, the end-to-end inference latency calculation formula in this paper is:

$$T_{total} = T_{enc} + \max(T_{sec}, T_{pub}) + T_{fuse} + T_{dec} \tag{7}$$

where:

- $T_{total}$is the total physical end-to-end inference latency;
- $T_{enc}$is the time consumed by the client for TFHE encryption of private data X_{sec}locally;
- $T_{sec}$is the time consumed by the server for TFHE ciphertext network evaluation of private data;
- $T_{pub}$is the time consumed by the cloud for plaintext network evaluation of non-private data X_{pub}in the secure black-box;
- $T_{fuse}$is the time consumed for homomorphic fusion of the two types of data;
- $T_{dec}$is the time consumed by the client to decrypt the final fused ciphertext H_{fuse}using the private key.

5.3 Results Comparison and Analysis

The primary challenge in fully homomorphic encryption inference is the nonlinear truncation of activation functions. Figure 4 shows the impact of different algorithms on the final model accuracy.

As shown in Figure 4, AutoFHE uses mixed-degree polynomials to approximate the ReLU function. Regardless of how the polynomial degree is increased, a continuous smooth polynomial curve can never perfectly fit the ReLU activation function. This irreversible approximation error accumulates layer by layer throughout the network, limiting the highest achievable accuracy to only 93.87%. In contrast, the proposed system innovatively utilizes Boolean logic control gates in the TFHE scheme during algorithm design, using the sign bit of the ciphertext two's complement as the control signal to achieve 100% zero-error truncation in the ciphertext domain. This fundamental difference allows the system to be almost completely free from any approximation noise interference, achieving a model accuracy of 96.14%, which exceeds the AutoFHE scheme by 2.27 percentage points.

As can be seen from Figure 5, traditional pure-ciphertext optimization schemes adopt the approach of placing all input data into the encryption domain. A large amount of meaningless non-private data in the original input significantly increases the burden of bootstrapping operations, resulting in persistently high total latency for such schemes. Among them, the best-performing optimization scheme (Co-designing) has a latency of 3100 ms, while AutoFHE and REDsec reach as high as 3500 ms and 4200 ms, respectively. The proposed system classifies data into private and non-private categories, sending only the data containing private information (X_{sec}) into the TFHE ciphertext domain, while performing plaintext computation on the majority

of non-private data (X_{pub}). Ultimately, the end-to-end total latency, including the homomorphic fusion time (T_{fuse}), is significantly compressed to only 970 ms.

Figure 4: Accuracy Comparison

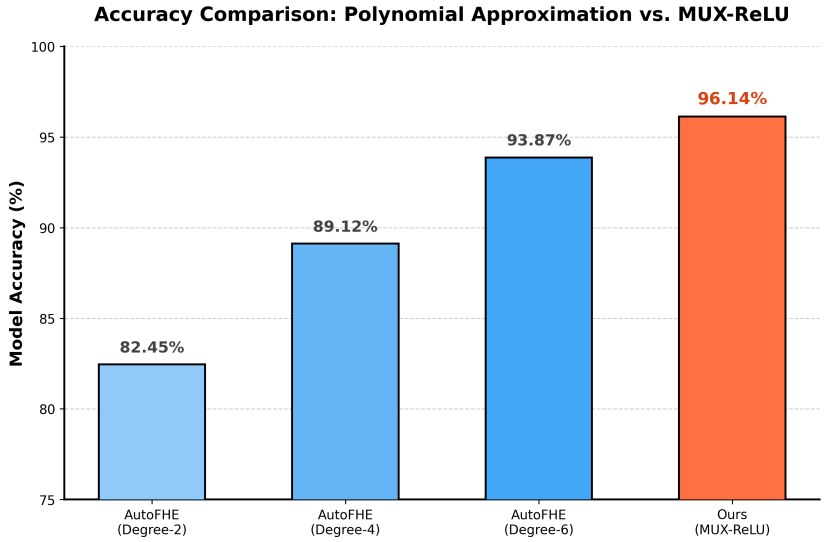


Figure 5: Latency Comparison

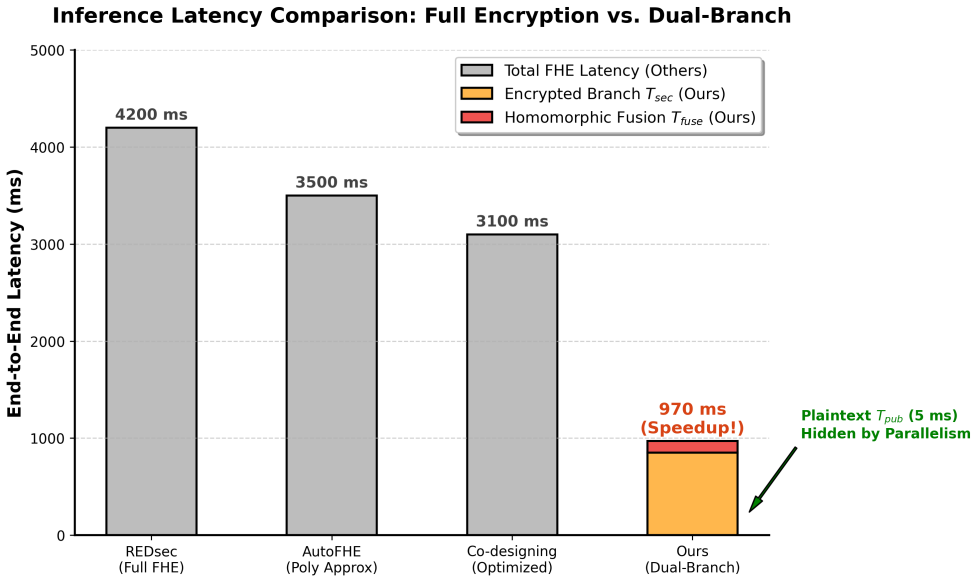


Table 2: Comprehensive Performance Comparison of Privacy-Preserving Deep Learning Systems

Evaluation Scheme	Encryption Mechanism	Nonlinear Activation Strategy	Test Set Accuracy	Cloud Inference Total Latency
AutoFHE	CKKS / BGV	Mixed-degree Polynomial Approximation	93.87%	≈ 3500 ms
REDsec	TFHE	Lookup Table Method	95.21%	≈ 4200 ms
Co-designing Activation	TFHE	Co-optimized Low-degree Polynomial	94.10%	≈ 3100 ms
Ours	TFHE + TEE	MUX Physical Zero-Error Truncation	96.14%	970 ms

As shown in Table 2, AutoFHE and the Co-designing Activation scheme reduce latency through polynomial approximation, but this causes their model accuracy to decrease, hovering around 93%–94%. Although REDsec maintains relatively high accuracy, the large volume of non-private data results in a latency as high as 4200 ms. In comparison, the proposed system achieves an accuracy of 96.14% through the dual-

path shunting mechanism and MUX logic gates, while dramatically compressing the end-to-end latency to 970 ms. This demonstrates the performance advantages of the proposed system.

5.4 Security Analysis

In dynamic scenarios with massive data, deep learning systems not only need to ensure low latency and high accuracy but also require certain robustness against attacks. Traditional encryption schemes often overlook the consumption of system computing resources by malicious traffic. This paper verifies the security advantages of the proposed dual-path heterogeneous scheme from four dimensions—man-in-the-middle attack [14], flooding attack [15], distributed denial-of-service (DDoS) attack [16], and jailbreak attacks against large models—by comparing it with the four schemes mentioned above.

First, the proposed system is compared with the other three schemes under complex attack scenarios. The specific evaluation results are shown in Table 3.

Table 3: Comparison of Anti-Attack Capabilities with State-of-the-Art FHE Privacy-Preserving Schemes

Defense Scheme	Resist MITM Attack	Flooding Attack	DDoS Attack	Jailbreak Attack
AutoFHE	Yes	Easily Blocked (Polynomial Overload)	System Paralysis (DoS)	Blind Computation (No Identification)
REDsec	Yes	Easily Blocked (Full Ciphertext Overload)	System Paralysis (DoS)	Blind Computation (No Identification)
Co-designing	Yes	Easily Blocked (Computation Bottleneck)	System Paralysis (DoS)	Blind Computation (No Identification)
Ours	Yes	Strong Resistance (Hash Pre-check & Discard)	Strong Resistance (Dual-path Compute Isolation)	Strong Resistance (TEE Logic Verification)

As shown in Table 3, all four schemes employ homomorphic encryption mechanisms and can effectively resist man-in-the-middle attacks based on traffic eavesdropping. However, when facing flooding attacks and DDoS attacks, existing literature schemes expose problems: schemes such as AutoFHE lack a data classification mechanism and choose to place all data into the encryption domain. If an attacker sends a large number of forged ciphertext requests, the server will perform computationally expensive bootstrapping operations on these data, consuming substantial CPU resources and causing the encryption needs of legitimate users to be ignored. In contrast, the proposed system performs a hash-based integrity check before data enters the secure black-box. Forged requests are identified and discarded before triggering ciphertext computation. At the same time, the dual-path hierarchical mechanism reduces the burden of encryption algorithms, enabling the system to achieve higher concurrent throughput. Additionally, traditional schemes cannot recognize jailbreak attacks and can only blindly process incoming illegal data. The secure black-box designed in this paper can perform behavioral feature checks on the data, effectively preventing illegal data from damaging the model.

6. Conclusions

To address the issues of data privacy leakage and server vulnerability in deep learning frameworks under distributed and dynamic environments, this paper proposes a dual-path hierarchical privacy-preserving inference method oriented toward dynamic and complex environments. First, a dual-path hierarchical preprocessing mechanism based on private data is designed. The input data is divided into private and non-private categories according to the importance of information. Private data is imported into the TFHE encryption domain for homomorphic encryption and computation, while non-private data is placed into a secure black-box for plaintext processing. Second, a MUX-based logic gate truncation mechanism is designed. By leveraging the bit-wise Boolean operations supported by the TFHE scheme, the ReLU activation function is accurately simulated, significantly improving model accuracy. Third, a secure black-box processing mechanism for non-private data is designed to protect such data. Finally, a cross-domain feature-level heterogeneous data fusion mechanism is constructed, enabling the secure fusion of the two types of data in an untrusted cloud environment.

Compared with existing state-of-the-art privacy-preserving frameworks, the proposed scheme improves end-to-end inference speed by at least 68.7%, achieves a model prediction accuracy of 96.14%, and realizes an order-of-magnitude leap in system concurrent throughput under DDoS attack defense. Although the proposed scheme has achieved remarkable results in balancing privacy computing efficiency and defending against DDoS attacks, there remains room for further exploration when facing even larger and more variable complex scenarios in the future. For example, future research can focus on extending the cross-domain heterogeneous concept presented in this paper from CNN models to large-scale Transformer architectures, and exploring the integration of encryption algorithms with ultra-large-scale artificial intelligence. In response to more complex future network confrontations, reinforcement learning can be introduced into the system, enabling it to dynamically adjust the data segmentation ratio according to current network security risks, so as to achieve an optimal balance between model precision and accuracy.

References

[1] Yang Li, Ruinong Wang, Yuanzheng Li, Meng Zhang, and Chao Long. Wind power forecasting considering data privacy protection: A federated deep reinforcement learning approach. Applied Energy, 329:120291, 2023.

[2] Sara Quach, Park Thaichon, Kelly D Martin, Scott Weaven, and Robert W Palmatier. Digital technologies: tensions in privacy and data. Journal of the academy of marketing science, 50(6):1299–1323, 2022.

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

[4] Kundan Munjal and Rekha Bhatia. A systematic review of homomorphic encryption and its contributions in healthcare industry. Complex & Intelligent Systems, 9(4):3759–3786, 2023.

[5] Alexander Viand, Christian Knabenhans, and Anwar Hithnawi. Verifiable fully homomorphic encryption. arXiv preprint arXiv:2301.07041, 2023.

[6] Wei Ao and Vishnu Naresh Boddeti. AutoFHE: Automated adaption of CNNs for efficient evaluation over FHE. In 33rd USENIX Security Symposium (USENIX Security 24), pages 2173–2190, 2024.

[7] Ershat Arkin, Nurbiya Yadikar, Xuebin Xu, Alimjan Aysa, and Kurban Ubul. A survey: object detection methods from cnn to transformer. Multimedia Tools and Applications, 82(14):21353–21383, 2023.

[8] Lars Folkerts, Charles Gouert, and Nektarios Georgios Tsoutsos. REDsec: Running encrypted discretized neural networks in seconds. Cryptology ePrint Archive, 2021.

[9] Abdulrahman Diaa, Lucas Fenaux, Thomas Humphries, Marian Dietz, Faezeh Ebrahimianghazani, Bailey Kacsmar, Xinda Li, Nils Lukas, Rasoul Akhavan Mahdavi, Simon Oya, et al. Fast and private inference of deep neural networks by co-designing activation functions. In 33rd USENIX Security Symposium (USENIX Security 24), pages 2191–2208, 2024.

[10] Kai Shen, Junliang Guo, Xu Tan, Siliang Tang, Rui Wang, and Jiang Bian. A study on relu and softmax in transformer. arXiv preprint arXiv:2302.06461, 2023.

[11] Tim Martin and Frank Allgöwer. Data-driven system analysis of nonlinear systems using polynomial approximation. IEEE Transactions on Automatic Control, 69(7):4261–4274, 2023.

[12] Xiaowei Yu, Zhe Huang, Yao Xue, Lu Zhang, Li Wang, Tianming Liu, and Dajiang Zhu. Noisynn: Exploring the influence of information entropy change in learning systems. arXiv preprint arXiv:2309.10625, 2023.

[13] Umberto Garlando, Qi Wang, Oleksandr V Dobrovolskiy, Andrii V Chumak, and Fabrizio Riente. Numerical model for 32-bit magnonic ripple carry adder. IEEE Transactions on Emerging Topics in Computing, 11(3):679–688, 2023.

[14] Muhanna Ahmed Ali and Salah Alawi Hussein Al-Sharafi. Intrusion detection in iot networks using machine learning and deep learning approaches for mitm attack mitigation. Discover Internet of Things, 5(1):48, 2025.

[15] Dmytro Tymoshchuk, Oleh Yasniy, Mykola Mytnyk, Nataliya Zagorodna, and Vitaliy Tymoshchuk. Detection and classification of ddos flooding attacks by machine learning method. arXiv preprint arXiv:2412.18990, 2024.

[16] S Abiramasundari and V Ramaswamy. Distributed denial-of-service (ddos) attack detection using supervised machine learning algorithms. Scientific Reports, 15(1):13098, 2025.

Funding

This research received no external funding.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgment

This paper is an output of the science project.

Copyrights