2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

Shadow Model Craft: An Efficient Framework for Privacy-Preserving Inference

Meijuan Li, Yidong wang, Ziwen wei and Zhe Cui(✉)

[1] Chengdu Institute of Computer Applications, Chinese Academy of Sciences, Chengdu, China, 610041
[2] University of Chinese Academy of Sciences, Beijing, China, 101408
limeijuan20@mails.ucas.ac.cn

Abstract. Privacy-preserving inference (PPI) has become a critical demand in Machine Learning as a Service (MLaaS), where both user inputs and model parameters and architecture are sensitive assets. Existing cryptographic-based approaches, such as those relying on homomorphic encryption or secure multi-party computation (MPC), often suffer from substantial computational and communication overhead, making them impractical for large-scale deep learning models. In this paper, we propose a novel and efficient framework that protects model and input privacy while significantly improving inference efficiency. The core of our approach is Shadow Model Craft, a structural model decomposition strategy inspired by secret sharing. Instead of encrypting model parameters, we distill the original model into multiple lightweight shadow models with disjoint functionality and distribute them across non-colluding servers. Each server performs inference over secret-shared inputs using plaintext model fragments, thus eliminating the need for encrypted model parameters. Our design allows local execution of linear operations, further reducing inference latency. Experiments on CIFAR-10 and ImageNet demonstrate that our framework achieves strong privacy guarantees with up to 90% model compression and over remarkable speedup compared to other sota works, while maintaining competitive inference accuracy. This work offers a practical and scalable solution for secure deep learning inference in real-world deployments.

Keywords: Privacy-Preserving Inference,Machine Learning as a Service (MLaaS), Multi-Party Computation.

1 Introduction

1.1 A Subsection Sample

Machine Learning as a Service (MLaaS) [1] has rapidly gained popularity, with cloud servers such as Amazon and Google enabling model providers to deploy neural networks in the cloud and deliver inference services to customers. By offloading storage and computing management to cloud platforms, MLaaS significantly reduces service delivery costs for vendors, and also simplifies customers' access to advanced AI capabilities. However, this business model raises significant privacy concerns, as both user data and critical model information are highly valuable and sensitive. For example, some sensitive input data—including facial images, financial records, or medical scans—must be protected from unauthorized access. Strict regulations, such as the [2] and China's Data Security Law [3], restrict how such data can be used, making direct cloud-based inference potentially risky. In addition, deployed models constitute valuable intellectual property for providers. To achieve high-performance neural networks, providers must invest heavily in curated datasets, intensive computation, and repeated optimization. Disclosure of critical model details can undermine proprietary value and expose providers to legal or commercial threats.

There is a growing need to protect the privacy of both input data and models in MLaaS environments. Many studies have explored cryptographic techniques to address this challenge, particularly in the context of convolutional neural networks (CNNs) [4]. These approaches commonly rely on cryptographic primitives such as homomorphic encryption [5] and multi-party computation (MPC) [6], with MPC generally offering better efficiency than HE in practice. Several representative systems, including GryptGPU [7] CrypTFlow [8] and [9], leverage MPC-based secret sharing to protect model parameters and user input. In these frameworks, model providers and clients act as separate data sources, while the cloud platform functions as computing party. This solution theoretically safeguards model parameters and user inputs, ensuring that only data sources have access to them. However, it offers limited protection for the model's architecture (including the number, types, and sizes of layers). This architectural design is considered proprietary and confidential [10]. Its disclosure may lead to significant risks for both intellectual property and commercial interests. Moreover, cryptographic methods inherently involve high computational complexity and communication overhead. These challenges are particularly pronounced when dealing with large-scale models containing billions of parameters in real-time applications. For example, state-of-the-art MPC solutions [7] take approximately 9 seconds to process a single image on a ResNet-50 model [11].

In this paper, we propose a novel and efficient framework for the MLaaS paradigm to mitigate security risks while maintaining inference accuracy. Our approach provides comprehensive privacy protection for both models and input/output data. Unlike existing cryptographic-based methods that uniformly apply expensive encryption techniques to both inputs and model parameters, our framework adopts customized protection strategies based on the nature of each component. Fully encrypting both the model and input significantly degrades inference performance, making such methods impractical in real-world scenarios. In contrast, our approach performs inference on a plaintext model, while preserving model confidentiality through decomposition and distributed deployment. Consequently, no single party involved in the process can independently access the model's function to reconstruct it's parameters or architecture.Our main contributions are summarized as follows:

1. **Efficient Privacy-Preserving Inference.** We present an efficitive privacy-preserving inference framework specifically designed for convolutional neural networks (CNNs), aiming to safeguard both user data and the proprietary aspects of

the model, including its parameters and architecture. Compared to existing privacy-preserving inference (PPI) approaches, our method significantly reduces inference latency while maintaining privacy guarantees.

2. **Shadow Model Craft.** We introduce a novel strategy inspired by secret sharing to structurally protect the model. By fragmenting the model's predictive behavior across multiple lightweight shadow models, this approach allows inference to be performed using plaintext models while ensuring that no single party can reconstruct the original model. The method also reduces the parameter size by up to 90%, greatly improving deployment and computation efficiency.
3. **Extensive Evaluation.** We conduct comprehensive experiments on CIFAR-10 and ImageNet using multiple CNN architectures. Results show that our framework achieves high inference accuracy and efficiency while significantly reducing the risk of model leakage during inference.

2 Related work

Privacy-preserving inference (PPI) has been extensively explored in recent years. One prominent line of work leverages cryptographic primitives, including differential privacy (DP) [12], homomorphic encryption (HE) [13], and multi-party computation (MPC) [14]. State-of-the-art systems such as SecureML [15], CrypTFlow2 [16], SecureNN [17], SSNet [18] and ABY3 [19] apply these techniques to encrypt both model parameters and user inputs, enabling secure computation via homomorphic operations or secret-sharing protocols.

Some studies seek to reduce inference latency by modifying model structures in PPI. Techniques such as minimal splitting [20], weight scaling [21], and matrix permutation [22] aim to construct MPC-friendly or privacy-preserving model variants. Ditto [23] proposes a quantization-aware distillation approach tailored to MPC protocols, while Private Model Compression via Knowledge Distillation explores distillation as a means to compress large models into lightweight student models, thereby improving inference efficiency. While knowledge distillation is traditionally used for model compression [24], it can also serve as a privacy-enhancing mechanism: by training student models to mimic the outputs of a private teacher model, the teacher model can remain undisclosed. However, the distilled student model often encapsulates a significant portion of the teacher's behavior and thus may itself become a valuable proprietary asset.

Compared to the above approaches, this work takes a fundamentally different direction by adopting knowledge distillation as the core mechanism for privacy-preserving inference. We introduce Shadow Model Craft, a strategy that fragments the predictive behavior of the original model across multiple lightweight shadow models. This design not only compresses the original model to improve inference efficiency, but also enables the model to be distributed across multiple servers for parallel execution. Most importantly, since each shadow model is functionally incomplete and structurally distinct from the original, model parameters can be transmitted in plaintext without compromising confidentiality. As a result, the inference protocol only needs to encrypt the input, allowing linear operations such as convolution and fully connected layers to be computed locally by each server. This significantly reduces inference latency compared to fully encrypted inference schemes, making it suitable for practical secure inference scenarios that demand high efficiency and low response time.

3 Preliminaries

3.1 Secret Sharing

To enable secure computation, we adopt a replicated secret sharing scheme designed for 3PC settings. All arithmetic operations are performed over the ring $\mathbb{Z}_n$, where $n = 2^{64}$ in our implementation. a private value x is encoded as a triple of additive shares such that $x = x_1 + x_2 + x_3 \bmod n$. Each party receives two out of the three shares in an overlapping pattern, ensuring that no single party can reconstruct the original value independently. For example, if the shares are (x_1, x_2, x_3), then party P_1 holds (x_1, x_2), P_2 holds(x_2, x_3), and P_3holds (x_3, x_1). and we denote that. This 2-out-of-3 sharing strategy offers two key benefits: it preserves data confidentiality under a semi-honest model, and it allows for efficient local computation with minimal communication overhead. We refer to this representation as $[\![x]\!]$, which satisfies REC($[\![x]\!]) = x$.

3.2 MPC Arithmetic

Linear components of CNNs, including convolution and dense layers, can be implemented efficiently using the aforementioned primitives. Let(a, b) be public constants and $[\![x]\!], [\![y]\!]$ be secret-shared values under replicated secret sharing. Operations such as $[\![z]\!] = a\ [\![x]\!] + b[\![y]\!]$ only involve addition and scalar multiplication by public constants, and can be performed locally by each party. For example, given share triplets (x_1, x_2, x_3) and (y_1, y_2, y_3), the resulting share of $(ax + by)$ is locally computed as $(ax_1 + by_1, ax_2 + by_2, ax_3 + by_3)$. In contrast, multiplying two secret-shared values requires interaction among parties. To compute $[\![x \cdot y]\!]$, each party locally expands the dot product $(x_1 + x_2 + x_3) \cdot (y_1 + y_2 + y_3) \bmod n$, which is split into three partial terms $(z_1 + z_2 + z_3) \bmod n$. Each term is computed as $z_i = x_i y_i + x_i y_{i+1} + x_{i+1} y_i) \bmod n$. These local products are then re-shared among the parties to obtain a fresh secret sharing of the result. In our implementation, we only perform scalar multiplications between secret-shared values and public constants, and do not involve multiplications between two secret-shared values.

4 Methodologies

4.1 Overview

The primary objective of this work is to propose an efficient and secure private inference scheme for the MLaaS paradigm, addressing the critical challenge of model extraction attacks. Our scheme effectively protects both the model's privacy (parameters and architecture) and client information (inputs and outputs) while maintaining high inference performance. Our proposed approach comprises two main components: Shadow Model Craft and a Secure Multi-party Inference Framework.

In the Shadow Model Craft, we introduce a novel scheme that transforms the original private model into an ensemble of shadow models. The correct inference result can only be obtained by aggregating the outputs from the entire ensemble. Secure multi-party inference Framework is based Three-party computation (3PC) setting, which is widely adopted as it offers stronger security guarantees than 2PC protocols while being more efficient than general n-party constructions. By distributing the shadow models across three computing entities and executing an MPC protocol in online inference stage, our method meets the required security guarantees and get higher performance. Detailed descriptions of each stage are provided in the following sections.

4.2 System model

We consider three types of entities in our framework:

- **Provider:** Owns a trained convolutional neural network (CNN) and deploys it across three cloud servers. The provider offers inference services while ensuring that the model remains confidential.
- **Servers:** A group of three non-colluding cloud servers that collaboratively execute the inference protocol through mutual interaction.
- **Client:** Submits input data to the cloud and receives the final prediction result.

Threat model. Similar to previous work, we assume a classical outsourced MPC paradigm. In this setting, providers and clients each submit their confidential data (model's parameters and architecture, inference queries and results) to non-colluding servers. These parties expect that their secrets remain secure and are not exposed to any unauthorized entity during computation. Inference results are also treated as sensitive, as repeated access can enable black-box attacks that extract the provider's model. The servers jointly run a MPC protocol. They perform computations over secret-shared data and return the secret-shared result only to the client. No server should learn anything about the confidential data. We assume a semi-honest adversary, all parties follow the protocol but may try to infer extra information from the messages they observe.

4.3 Shadow Model Craft

Design Motivation. We are inspired by the principle of distributed trust and the concept of secret sharing. Rather than splitting confidential model parameters into multiple shares, we aim to protect the model through structural decomposition. The central idea of **Shadow Model Craft** is to fragment the model's predictive behavior into multiple components, such that the correct inference result can only be obtained when the outputs of all components are available.

During online inference, the original model is securely held by the provider, while different servers control separate components. The model is not merely partitioned but restructured through ensemble knowledge distillation. Each resulting component is functionally incomplete and architecturally distinct from the original, preventing any single party from inferring the original model, even with full access to their assigned part.

After decomposition, all components can remain in plaintext, enabling efficient and privacy-preserving inference. As demonstrated empirically, this structure-based protection significantly reduces the overhead compared to traditional MPC approaches.

Ensemble Knowledge Distillation. As shown in Algorithm , the private model acts as a teacher to train an ensemble of lightweight shadow models $\{M_1, M_2, \dots, M_n\}$.

The teacher first processes the training dataset to generate soft labels, which contain richer information than hard labels and help guide the student models to approximate its output distribution more accurately.

Each shadow model is initialized independently and adopts a lightweight architecture that differs from the teacher. To enhance privacy, we also apply label-space slicing during initialization: the overall set of class labels is partitioned into disjoint subsets, and each shadow model is trained to predict only the classes within its assigned subset. As a result, no single model has access to the full label space, which limits its ability to approximate the teacher model independently.

Every shadow model is assigned a fixed aggregation coefficient c_i, where $\sum_{i=1}^{n} c_i = 1$. These coefficients determine the relative contribution of each model to the final ensemble output, similar to assigning voting weights in a committee.

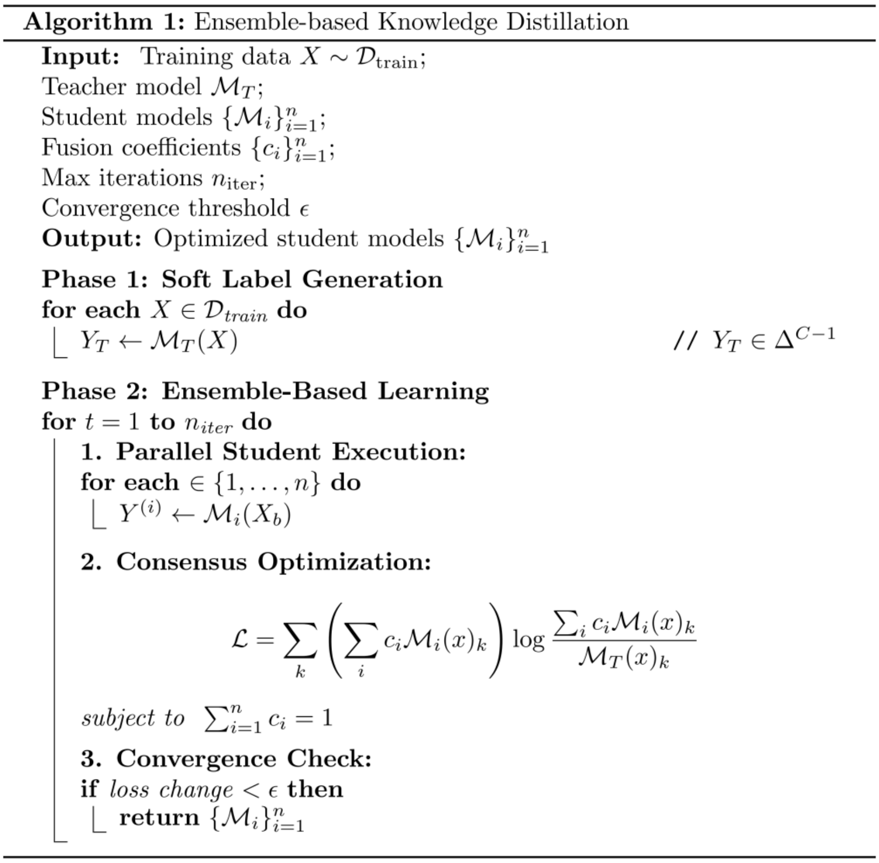
All shadow models are trained simultaneously over multiple rounds of distillation until they converge on the soft labels. During training, each model independently processes the same input samples. Their predictions are aggregated using a weighted sum based on the coefficients $\{c_i\}$, resulting in the final ensemble output:

$$M(x) = \sum_{i=1}^{n} c_i \, M_i(x)$$

The ensemble output is then compared to the teacher model prediction using the Kullback–Leibler (KL) divergence. The distillation loss is defined as:

$$\mathcal{L} = \sum_k \left(\sum_i c_i \, M_i(x)_k \right) \log \frac{\sum_i c_i \, M_i(x)_k}{M_{\text{teacher}}(x)_k}$$

This objective allows the ensemble to closely mimic the teacher's behavior, while ensuring that each individual shadow model remains incomplete and unable to make accurate predictions on its own. The provider can adjust both the number of models n and the aggregation $\{c_i\}_{i=1}^{n}$ to to flexibly balance inference performance and privacy at different levels. A larger number of models generally improves the overall accuracy of the ensemble, while also enhancing privacy by distributing knowledge more sparsely across components. By adjusting the aggregation weights$\{c_i\}_{i=1}^{n}$, the provider can further control the contribution of each model to the final output. For instance, assigning equal weights to all c_i gives each model the same influence during inference, which increases privacy by preventing any single model from dominating the prediction.



```
Algorithm 1: Ensemble-based Knowledge Distillation
  Input: Training data $X \sim \mathcal{D}_{\text{train}}$;
  Teacher model $\mathcal{M}_T$;
  Student models $\{\mathcal{M}_i\}_{i=1}^n$;
  Fusion coefficients $\{c_i\}_{i=1}^n$;
  Max iterations $n_{\text{iter}}$;
  Convergence threshold $\epsilon$
  Output: Optimized student models $\{\mathcal{M}_i\}_{i=1}^n$
  Phase 1: Soft Label Generation
  for each $X \in \mathcal{D}_{train}$ do
      $Y_T \leftarrow \mathcal{M}_T(X)$                          // $Y_T \in \Delta^{C-1}$
  Phase 2: Ensemble-Based Learning
  for $t = 1$ to $n_{iter}$ do
      1. Parallel Student Execution:
      for each $\in \{1, \dots, n\}$ do
          $Y^{(i)} \leftarrow \mathcal{M}_i(X_b)$
      2. Consensus Optimization:
          $\mathcal{L} = \sum_k \left( \sum_i c_i \mathcal{M}_i(x)_k \right) \log \frac{\sum_i c_i \mathcal{M}_i(x)_k}{\mathcal{M}_T(x)_k}$
      subject to $\sum_{i=1}^n c_i = 1$
      3. Convergence Check:
      if loss change $< \epsilon$ then
          return $\{\mathcal{M}_i\}_{i=1}^n$
```

Deployment and Security Implications. All trained shadow models are deployed across distinct computing parties. Since the correct prediction can only be reconstructed when all models contribute, any subset of models reveals only partial and insufficient information. This design significantly protects the confidentiality of original model, and also reduces the risk of model extraction by any single party.

4.4 Secure Multi-party Inference

We now present the full secure inference process, based on the MPC protocol and the shadow model structure. Although this work adopts a three-server configuration, the underlying design is readily extensible to support more parties with minimal modifications.

Secure Module. In neural networks, linear layers transform input representations into new dimensions, typically through dense or convolutional operations. In our secure inference setting, these computations are represented as $\llbracket y \rrbracket =$

$\mathrm{Dense}([\![x]\!], w)$ and $[\![y]\!] = \mathrm{Conv2D}([\![x]\!], w)$, where w denotes plaintext model weights. Since these operations involve only linear combinations between secret-shared inputs and public constants, they can be performed locally by each party without requiring interaction.

Beyond linear layers, CNNs also incorporate non-linear functions such as ReLU, comparisons, and fixed-point truncation. These are treated as black-box primitives provided by existing MPC frameworks [25] and are used directly in our implementation without modification.

Offline Phase. Using the confidential model$M$and a selected set of aggregation coefficients $\{c_1, c_2, c_3\}$, the provider invokes to generate three shadow models$\{M_1, M_2, M_3\}$.

To emulate replicated secret sharing, the models are distributed as follows: $P_1 \rightarrow (M_1, M_2)$, $P_2 \rightarrow (M_2, M_3)$, and $P_3 \rightarrow (M_1, M_3)$. The aggregation coefficients are sent to the client for later reconstruction. No single server can access the complete predictive behavior of M, as all three components and their corresponding coefficients are required to recover the final result.

Protocol 1: Shadow Model Generation

Input: Confidential model $\mathcal{M}$; aggregation coefficients $\{c_1, c_2, c_3\}$ with $\sum_{i=1}^{3} c_i = 1$.
Output: Shadow models $\{\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3\}$.
Steps:

1. Use Algorithm 1 to train shadow models $\{\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3\}$ from $\mathcal{M}$ by minimizing the KL divergence between $\sum_{i=1}^{3} c_i \mathcal{M}_i(x)$ and $\mathcal{M}(x)$.
2. Output the trained shadow models $\{\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3\}$ and the coefficients $\{c_1, c_2, c_3\}$.

Online Phase. Built upon replicated secret primitives as introduced in Section 2. In particular, the client first applies replicated secret sharing to split its input x into three shares $\{x_1, x_2, x_3\}$, and distributes them to the computing parties as follows: $P_1 \rightarrow (x_1, x_2)$, $P_2 \rightarrow (x_2, x_3)$, $P_3 \rightarrow (x_1, x_3)$.

Each server uses the secure module to evaluate its assigned shadow models on the received input shares. As previously discussed, all computations in linear module are performed locally by each server, without requiring communication. Communication is only required for non-linear operations such as ReLU and fixed-point truncation. For example, to jointly compute the output share of $M_1(x)$, serversP_1 and P_3 must engage in a secure protocol during the non-linear layers. Similarly, P_1 communicates with P_2 to obtain its portion of the output from $M_2(x)$.

After the joint computation phase, each server holds partial outputs as follows:

- P_1 holds $\{M_1(x_1), M_1(x_2), M_2(x_1), M_2(x_2)\}$
- P_2 holds $\{M_2(x_2), M_2(x_3), M_3(x_2), M_3(x_3)\}$
- P_3 holds $\{M_1(x_1), M_1(x_3), M_3(x_1), M_3(x_3)\}$

These partial results are then sent back to the client, who performs reconstruction and aggregation to obtain the final prediction. The output of each shadow model is shared secret between the parties, so the client must first reconstruct the full output of each model by summing the received shares. For example, to reconstruct $M_1(x)$, the client computes $M_1(x) = M_1(x_1) + M_1(x_2) + M_1(x_3)$. After obtaining $M_1(x), M_2(x)$ and $M_3(x)$, the final prediction is computed by weighted aggregation: $M(x) = c_1 \cdot M_1(x) + c_2 \cdot M_2(x) + c_3 \cdot M_3(x)$.

Protocol 2: Secure Online Inference

Input:

- Client input x
- Shadow models $\{M_1, M_2, M_3\}$ held by distributed computing parties
- Aggregation coefficients $\{c_1, c_2, c_3\}$ such that $\sum_{i=1}^{3} c_i = 1$

Output: Final prediction $M(x) = \sum_{i=1}^{3} c_i M_i(x)$ is computed collaboratively and revealed only to the client.

5 Experiments and Results

5.1 Setup

Our experiments are built upon the CrypTen [25] and CryptGPU [26] framework, whose data security has been theoretically verified within the framework. And we conducted in a local area network (LAN) setting. Each device is equipped with one NVIDIA RTX 3090 GPU with 24 GB of memory. We evaluate our method on two widely used benchmark datasets, CIFAR-10 and ImageNet, using ResNet-50, ResNet-101, ResNet-152, VGG-16, and VGG-19 as orginal models. All parties run under Ubuntu 20.04 with PyTorch 2.0.1 installed.

5.2 Inference Analysis

We set the number of shadow models to 3, with each model assigned an equal weight of $c_i = \frac{1}{3}$. To verify the generality of our method, each shadow model is constructed as a lightweight network composed of randomly selected residual blocks. We begin by evaluating the inference accuracy of the ensemble of shadow models. As shown in **Table 1.** Comparison of confidential models and shadow models (* indicates shadow models. ^{1} denotes the evaluation results on CIFAR-10, and ^{2} corresponds to those on ImageNet)., the ensemble can maintain the inference performance of the original teacher model with only a slight drop in precision. Notably, model distillation results in a substantial reduction in parameter size—achieving up to 90% compression compared to the original model. This significantly reduces both storage requirements and the computation overhead in the MPC setting. Therefore, in terms of inference performance, the student ensemble can effectively replace the original model.

Table 1. Comparison of confidential models and shadow models (* indicates shadow models. [1] denotes the evaluation results on CIFAR-10, and [2] corresponds to those on ImageNet).

Network	Params (M)	Test Acc.[1] (%)	Test Acc.[2] (%)
ResNet-50	25.6	94.1	76.2
ResNet-50*	2.4	93.6 ± 0.12	73.4
ResNet-101	44.6	95.1	77.4
ResNet-101*	3.1	94.8 ± 0.08	74.2
ResNet-152	60.2	95.6	78.6
ResNet-152*	2.8	94.6 ± 0.21	74.9
VGG-16	132	93.2	71.8
VGG-16*	5.1	92.7 ± 0.33	70.5
VGG-19	143	94.2	72.6
VGG-19*	4.4	93.8 ± 0.14	71.2

Table 2. Test accuracy under different numbers of Shadow Models

Network	1-Shadow Acc. (%)	2-Shadow Acc. (%)	3-Shadow Acc. (%)
ResNet-50	18.6	17.6	93.6
ResNet-101	12.5	17.1	94.8
ResNet-152	14.5	12.2	94.8
VGG-16	15.2	16.6	92.7
VGG-19	15.5	16.9	93.8

5.3 Security Analysis

We further conduct inference experiments on CIFAR-10 using either a single shadow model or any pair of shadow models. As shown in **Table 2**, when only one or two shadow models participate in the inference, the prediction accuracy is nearly indistinguishable from random guessing. This is due to the effect of our knowledge-slicing technique, which ensures that each student model learns only a partial subset of the teacher model's functionality. In contrast, when all three models are involved, the system achieves high prediction accuracy thanks to our ensemble distillation mechanism. These results demonstrate that in our 3PC inference setting, any two shadow models are deployed in plaintext on arbitrary servers, compromising any single server reveals no meaningful information about the inference functionality of the original secret model. As a result, model extraction attacks and other potential security breaches can be effectively prevented.

5.4 Efficiency Analysis

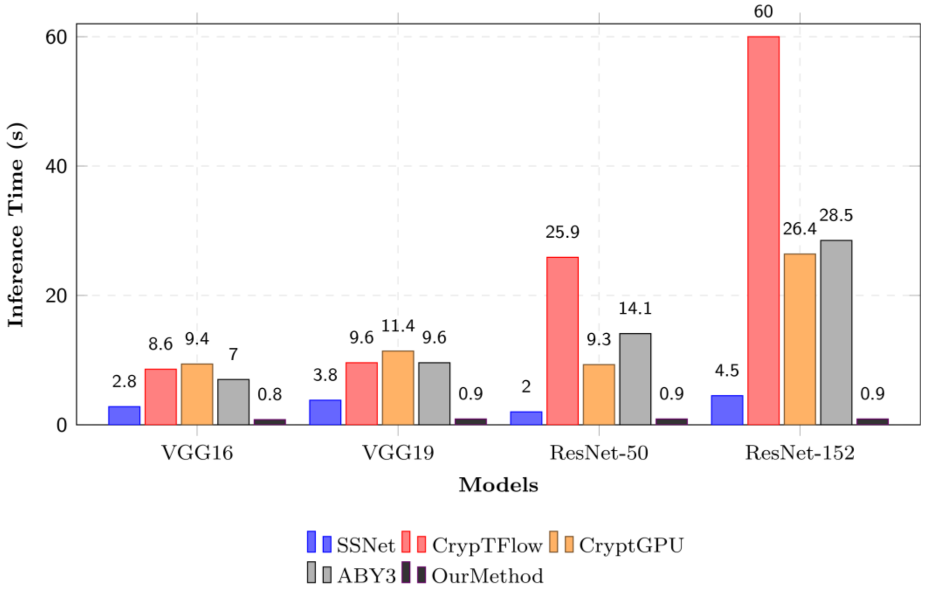


Fig .1. Inference time comparison across different models using SSNet [18], CrypTFlow [8], CryptGPU [26], ABY3 [19], and OurMethod.

As shown in

Table 1, our method significantly reduces inference time overhead by replacing the original large models with shadow models. These student models are randomly generated and have similar sizes, leading to nearly consistent inference times across different teacher networks after distillation. In other words, the inference cost of our method is independent of the size and structure of the original models. Furthermore, to evaluate the inference efficiency of our method, we compare it with several state-of-the-art secure inference approaches, as **Fig .1.** Inference time comparison across different models using SSNet [18], CrypTFlow [8], CryptGPU [26], ABY3 [19], and OurMethod. **Fig .1**, Since our method employs lightweight student models and does not require encryption of model parameters, it significantly reduces the time overhead of secure inference while still achieving state-of-the-art results.

5.5 Ablation Study

To understand the trade-off between model ensemble size and inference accuracy, we conduct an ablation study by varying the number of shadow models used in the inference. We investigate the impact of the number of shadow models n on inference accuracy. Specifically, we extract between 2 and 7 student models from a ResNet-50 and evaluate their ensemble inference accuracy under non-encrypted settings on CIFAR-10. As shown in **Fig. 2** , increasing the number of student models consistently improves the overall prediction accuracy. However, this accuracy gain comes at the cost of increased computational overhead, particularly during the online inference phase.

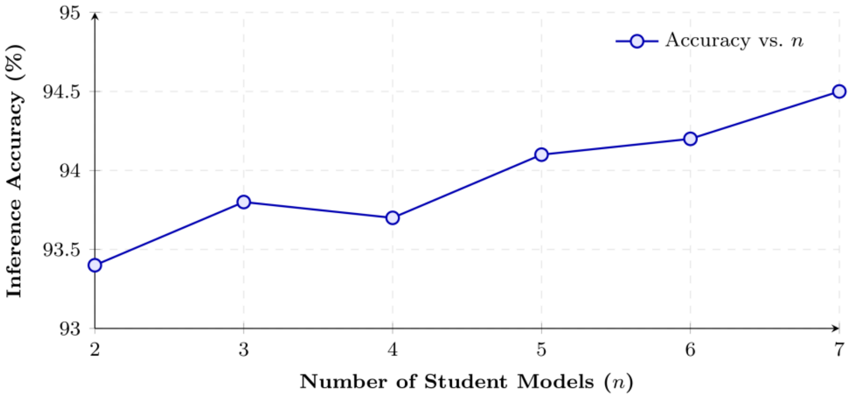


Fig. 2. Inference accuracy of ensembles with different numbers of student models on CIFAR-10

6 Conclusion

In this work, we present a novel privacy-preserving inference framework that effectively balances security, efficiency, and accuracy in the MLaaS setting. Unlike traditional cryptographic approaches that encrypt both model parameters and input， often resulting in significant inference latency, we introduce Shadow Model Craft, a structure-aware defense mechanism that protects model confidentiality without requiring parameter encryption. By decomposing the model's predictive behavior into multiple lightweight shadow models and distributing them across non-colluding servers, our method ensures that no single party can reconstruct or infer the original model. Meanwhile, the client-side input remains protected via secure multi-party computation. Experimental results on CIFAR-10 and ImageNet demonstrate that our framework significantly reduces inference latency while maintaining high prediction accuracy. Overall, our design provides a practical and scalable solution for CNN inference in real-world deployment environments.

References

1. Weng, Q., Xiao, W., Yu, Y., Wang, W., Wang, C., He, J., Li, Y., Zhang, L., Lin, W., Ding, Y.: MLaaS in the Wild: Workload Analysis and Scheduling in Large-Scale Heterogeneous GPU Clusters. In: 19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22), pp. 945–960 (2022)

2. Voigt, P., Von dem Bussche, A.: The EU General Data Protection Regulation (GDPR). 1st edn. Springer, Cham (2017)
3. Chen, J., Sun, J.: Understanding the Chinese Data Security Law. International Cybersecurity Law Review 2(2), 209–221 (2021)
4. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet Classification with Deep Convolutional Neural Networks. Adv. Neural Inf. Process. Syst. 25 (2012)
5. Joshi, B., Joshi, B., Mishra, A., Arya, V., Gupta, A.K., Peraković, D.: A Comparative Study of Privacy-Preserving Homomorphic Encryption Techniques in Cloud Computing. Int. J. Cloud Appl. Comput. 12(1), 1–11 (2022)
6. Zhang, X., Chen, C., Xie, Y., Chen, X., Zhang, J., Xiang, Y.: A Survey on Privacy Inference Attacks and Defenses in Cloud-Based Deep Neural Network. Comput. Stand. Interfaces 83, 103672 (2023)
7. Tan, S., Knott, B., Tian, Y., Wu, D.J.: CryptGPU: Fast Privacy-Preserving Machine Learning on the GPU. In: IEEE Symposium on Security and Privacy (SP), pp. 1021–1038 (2021)
8. Kumar, N., Rathee, M., Chandran, N., Gupta, D., Rastogi, A., Sharma, R.: CryptFlow: Secure TensorFlow Inference. In: IEEE Symposium on Security and Privacy (SP), pp. 336–353 (2020)
9. Wagh, S., Tople, S., Benhamouda, F., Kushilevitz, E., Mittal, P., Rabin, T.: Falcon: Honest-Majority Maliciously Secure Framework for Private Deep Learning. arXiv preprint arXiv:2004.02229 (2020)
10. Mann, Z.Á., Weinert, C., Chabal, D., Bos, J.W.: Towards Practical Secure Neural Network Inference: The Journey So Far and the Road Ahead. ACM Comput. Surv. 56(5), 1–37 (2023)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: IEEE Conf. on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
12. Blanco-Justicia, A., Sánchez, D., Domingo-Ferrer, J., Muralidhar, K.: A Critical Review on the Use (and Misuse) of Differential Privacy in Machine Learning. ACM Comput. Surv. 55(8), 1–16 (2022)
13. Fan, J., Vercauteren, F.: Somewhat Practical Fully Homomorphic Encryption. Cryptology ePrint Archive (2012)
14. Goldreich, O., Micali, S., Wigderson, A.: How to Play Any Mental Game, or a Completeness Theorem for Protocols with Honest Majority. In: Providing Sound Foundations for Cryptography, pp. 307–328 (2019)
15. Liu, F., Yu, Y.: Scalable Multi-Party Computation Protocols for Machine Learning in the Honest-Majority Setting. USENIX Security 1956 (1939)
16. Rathee, D., Rathee, M., Kumar, N., Chandran, N., Gupta, D., Rastogi, A., Sharma, R.: CryptFlow2: Practical 2-Party Secure Inference. In: ACM SIGSAC Conference on Computer and Communications Security, pp. 325–342 (2020)
17. Wagh, S., Gupta, D., Chandran, N.: SecureNN: 3-Party Secure Computation for Neural Network Training. Proc. Priv. Enhancing Technol. (2019)
18. Duan, S., Wang, C., Peng, H., Luo, Y., Wen, W., Ding, C., Xu, X.: SSNet: A Lightweight Multi-Party Computation Scheme for Practical Privacy-Preserving Machine Learning Service in the Cloud. arXiv preprint arXiv:2406.02629 (2024)
19. Mohassel, P., Rindal, P.: ABY3: A Mixed Protocol Framework for Machine Learning. In: ACM SIGSAC Conference on Computer and Communications Security, pp. 35–52 (2018)
20. Zhou, T., Luo, Y., Ren, S., Xu, X.: NNSplitter: An Active Defense Solution for DNN Model via Automated Weight Obfuscation. In: Int. Conf. on Machine Learning, pp. 42614–42624 (2023)

21. Shen, T., Qi, J., Jiang, J., Wang, X., Wen, S., Chen, X., Zhao, S., Wang, S., Chen, L., Luo, X., et al.: SOTER: Guarding Black-Box Inference for General Neural Networks at the Edge. In: USENIX ATC, pp. 723–738 (2022)
22. Sun, Z., Sun, R., Liu, C., Chowdhury, A.R., Lu, L., Jha, S.: ShadowNet: A Secure and Efficient On-Device Model Inference System for CNNs. In: IEEE Symposium on Security and Privacy (SP), pp. 1596–1612 (2023)
23. Wu, H., Fang, W., Zheng, Y., Ma, J., Tan, J., Wang, Y., Wang, L.: Ditto: Quantization-Aware Secure Inference of Transformers upon MPC. arXiv preprint arXiv:2405.05525 (2024)
24. Chen, P., Liu, S., Zhao, H., Jia, J.: Distilling Knowledge via Knowledge Review. In: IEEE/CVF Conf. on Computer Vision and Pattern Recognition, pp. 5008–5017 (2021)
25. Knott, B., Venkataraman, S., Hannun, A., Sengupta, S., Ibrahim, M., van der Maaten, L.: CrypTen: Secure Multi-Party Computation Meets Machine Learning. Adv. Neural Inf. Process. Syst. 34, 4961–4973 (2021)
26. Tan, S., Knott, B., Tian, Y., Wu, D.J.: CryptGPU: Fast Privacy-Preserving Machine Learning on the GPU. In: IEEE Symposium on Security and Privacy (SP), pp. 1021–1038 (2021)