*ECAI 2024*
*U. Endriss et al. (Eds.)*


*doi:10.3233/FAIA240477*

# TrustMIS: Trust-Enhanced Inference Framework for Medical Image Segmentation

**Fuyi Wang[a], Jinzhi Ouyang[b], Lei Pan[a], Leo Yu Zhang[c,*], Xiaoning Liu[d], Yanping Wang[e] and Robin Doss[a]**

[a]Deakin University, Australia
[b]Xiamen University, China
[c]Griffith University, Australia
[d]RMIT University, Australia
[e]University of Electronic Science and Technology of China, China

**Abstract.** Recent advancements in privacy-preserving deep learning (PPDL) enable artificial intelligence-assisted (AI-assisted) medical image diagnostics with privacy guarantees, addressing increasing concerns about data and model privacy. However, intensive studies are restricted to shallow and narrow neural networks (NNs) for simple service (e.g., disease prediction), leaving a gap in exploring diverse inferences. This paper proposes TrustMIS, a trust-enhanced inference framework for fast and private medical image segmentation (MIS) and prediction services. Based on two-party computation, TrustMIS introduces lightweight additive secret-sharing tools to safeguard medical records and NNs. Complementing existing PPDL schemes, we present a series of secure two-party interactive protocols for linear layers. Specifically, we optimize the secure matrix multiplication by reducing the number of expensive multiplication operations with the help of free-computation addition operations to enhance efficiency (bringing $1.15\times \sim 2.64\times$ savings in both time and communication costs). Furthermore, we customize a fresh secure transposed convolutional protocol for MIS-oriented NNs. A thorough theoretical analysis is provided to prove TrustMIS's correctness and security. We conduct experimental evaluations over two benchmark and four real-world medical datasets and compare them to state-of-the-art studies. The results demonstrate TrustMIS's superiority in efficiency and accuracy, improved by $1.1\times \sim 54.4\times$ speedup in secure disease prediction, and $5.56\% \uparrow \sim 11.7\% \uparrow$ accuracy in secure MIS.

## 1 Introduction

Recently, the rapid advancement of deep learning (DL) techniques has significantly impacted medical domains, ranging from drug discovery and clinical trials, personalized treatment, to medical diagnostics [23, 11]. Leveraging neural networks (NNs), enterprises provide various AI-assisted medical diagnostic services, enabling hospitals to enhance the speed and accuracy of medical decisions based on their medical records. However, alongside the proliferation of such services, there have been growing concerns about medical records security and privacy [6, 3, 4]. Medical records are inherently sensitive and hospitals may be hesitant to share sensitive patient data with service-provided enterprises. Additionally, NN models utilized in these services are considered valuable intellectual property and must always be kept confidential. To mitigate privacy concerns from the above

* Corresponding Author. Email: leo.zhang@griffith.edu.au

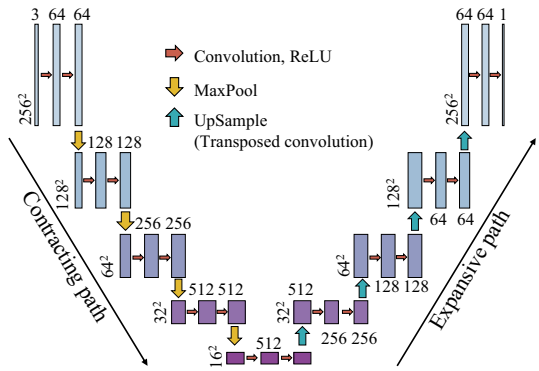


**Figure 1.** A sample of 5-depth U-Net architecture for MIS.

AI-assisted service scenarios, cryptographic techniques like differential privacy (DP) and homomorphic encryption (HE) are often integrated to create privacy-preserving deep learning (PPDL) schemes. However, DP leads to severe performance degradation in exchange for privacy [9], and pure HE imposes significant computational overhead, leading to impractical latency [13].

To overcome these limitations, multi-party computation (MPC) tools are employed to design promising PPDL solutions for secure inference over encrypted/secret-shared NN models and medical records. Notable MPC-based PPDL schemes include XONN [21], Cheetah [8], CrypTFlow2 [19], SiRnn [20], Bicoptor [25], and many others. These schemes primarily aim to bridge the efficiency and accuracy gaps between plaintext and ciphertext computation for *secure prediction*. Real-world AI-assisted medical diagnostic scenarios extend beyond prediction service to encompass applications like medical image segmentation (MIS). MIS plays a crucial role in accurately identifying and analyzing anatomical or pathological structures and enables the precise detection of anomalies, tumors, and abnormal regions within medical records. However, existing MPC-based PPDL schemes face two problems for MIS.

*Problem 1: The absence of secure protocols for the unique layers in MIS-oriented NNs.* Traditional NN models for secure disease prediction differ from the specific NN models (i.e., U-Nets) for MIS. For instance, U-Nets require unique transposed convolutional layers to upsample feature maps to match the original image size. Existing MPC-based PPDL schemes fail to provide protocols for these U-Nets' unique layers. *Problem 2: The absence of optimizing the*

*computation complexity of linear layers.* Existing schemes prioritize enhancing the performance of non-linear layers due to the limited number of linear layers in prediction models. Consequently, the optimization of linear layers is often overlooked [25, 14]. This oversight has particularly severe implications in MIS-oriented U-Nets, which involve an overwhelming amount of linear layers (convolutional and transposed convolutional layers), as illustrated in Figure 1.

It is challenging to devise secure and efficient protocols for these indispensable layers within U-Nets and generic linear layers. *Firstly*, these linear layers play a crucial role in feature extraction and segmented image reconstruction, any loss in precision during linear computations in the encrypted/secret-shared domain can directly impact the final segmentation results. Therefore, these designed secure layers must guarantee computational accuracy to reconstruct the segmented image accurately. *Secondly*, ensuring efficient computation is crucial to minimize inference time and enhance user experience, especially when dealing with large medical images. Thus, how to reduce the computational complexity without compromising computational accuracy is key when designing secure protocols for MIS.

In response to the above challenges, we design, implement, and evaluate TrustMIS, a lightweight and trust-enhanced inference framework for AI-assisted medical diagnostics. By combining the advancements from cryptography and mathematical realms, TrustMIS is a full-fledged cloud service framework that can support both generic NN and U-Net models, and provide high-performance secure inference. Specifically, TrustMIS falls in the secure two-party computation (2PC) paradigm to distribute secure inference among two non-colluding cloud servers, liberating end devices and model owners from being actively online for assistance. Over the secret-shared inputs, advancements in mathematical realms are leveraged to streamline both computational and communication processes of secure multiplication-related protocols. We then provide insights into how these protocols seamlessly apply to linear layers involved in various NN models, thereby enhancing overall efficiency. Additionally, by designing delicate secure transposed convolutional layers for U-Nets, TrustMIS improves efficiency performance and maintains accuracy in MIS. In a nutshell, our contributions are threefold.

- We introduce TrustMIS, a lightweight and trust-enhanced inference framework for medical image segmentation and prediction services. With the 2PC paradigm, TrustMIS employs lightweight additive secret-sharing techniques to distribute the input data and the well-trained NN model among two non-colluding servers, thereby protecting their privacy.
- We customize the novel secure matrix multiplication (MM) protocols `SecMaMult` for traditional linear layers harnessing the insights from cryptography and mathematics. For MIS-oriented models (i.e., U-Nets), we reformulate transposed convolutional layers through mathematical transformations. This allows for seamless integration with the proposed `SecMaMult`, facilitating efficient and secure transposed convolutional computation.
- We prove the correctness and security of TrustMIS by theoretical analysis. Extensive experiments on benchmark and healthcare datasets demonstrate that the newly developed protocols outperform existing PPDL counterparts in terms of efficiency, and TrustMIS performs comparably to unprotected MIS services.

## 2 Related Works

### *2.1 Privacy-Preserving Medical Image Segmentation*

Various studies use autoencoders to encode medical images to discard private information, achieving a balance between performance and privacy protection for MIS [12, 13, 26]. For instance, Mixup-privacy [13] executes secure MIS by training an encoder and then encodes target images by mixing them with reference patches (i.e., ground-truth) to guarantee the security of medical images. While an encoder may help reduce the risk of privacy leakage in certain scenarios, it remains highly sensitive to changes in the distribution of input images. Such sensitivity poses a risk to user identity privacy if the distribution shifts [9]. Therefore, achieving a higher level of privacy protection typically requires integrating cryptographic techniques. Jiang *et al*.[9] utilize a cryptographic technique, named differential privacy (DP), to safeguard user medical record privacy, but there exists a trade-off between privacy protection and performance. Specifically, under high privacy protection levels, such as using a security parameter $z = 0.7$ for prostate MRI segmentation, severe performance degradation occurs, with segmentation accuracy reaching only $59\%$. Nandakumar *et al*.provide robust security guarantees for medical records by using HE without accuracy degradation [18]. While the success of HE-based schemes is limited by the prohibitive and impractical computational and communication overhead. Even the computation of simple NNs is exceptionally slow, taking up to 30 minutes to process a single image segmentation service [13]. Thus, current research has yet to find an ideal solution that both ensures practical performance and protects privacy.

### *2.2 MPC-based Privacy-Preserving Deep Learning*

The widespread adoption of cloud-based DL has raised privacy concerns. To address this issue, extensive studies propose PPDL schemes operated by distinct servers from independent cloud providers to jointly provide secure AI-assisted services, like SecureML [17], GAZELLE [10], and XONN [21]. Despite differences in their detailed designs, these works require to employ heavy cryptographic tools, such as HE and garbled circuits, resulting in impractical latency during secure inference. Notably, secure ReLU layers contribute significantly to latency in secure inference. To address this, many recent works, including CrypTFlow2 [19], SiRnn [20], PAPI [2], PCNNCEC [22], Pio [24], and Bicoptor [25], focus on accelerating cryptographic computations for ReLU layers to enhance the efficiency of PPDL schemes. They leverage the strengths and mitigate the pitfalls of diverse cryptographic tools. For instance, CrypTFlow2 [19] introduces secure ReLU protocol SECRELU and secure max-pool protocol SECMAXPOOL by combining communication-saving HE and computation-saving oblivious transfer.

To overcome the performance bottleneck in PPDL and close the gap between cleartext and ciphertext inference, researchers realize the significance of enhancing the efficiency of linear layers (i.e., multiplication operations), as a typical NN model often involves millions or even billions of multiplication operations. Modern high-performance works such as Delphi [16], Cheetah [8], and FastSecNet [7] optimize secure computations for non-linear layers as well as the previously largely overlooked linear layers. Despite their usefulness, they only focus on the NN models for prediction and do not fully support modern NN models, such as U-Net, for semantic segmentation. Furthermore, their optimized linear protocols merely shift some computations offline without fundamentally reducing the computational complexity. Building on the advancements in full-developed secure non-linear layers (i.e., SECRELU and SECMAXPOOL) [19, 8], TrustMIS revisits the secure two-server (i.e., two-party) computations of linear layers and customizes fast protocols for modern NN models to provide more comprehensive secure medical diagnostics beyond prediction services.

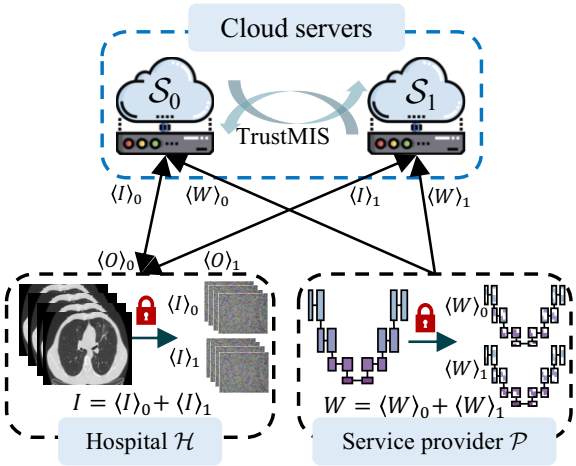


**Figure 2.** The system architecture.

## 3 System Overview

### 3.1 Architecture

TrustMIS targets a standard scenario of secure AI-assisted medical inference (including prediction and segmentation). As shown in Figure 2, TrustMIS comprises two servers $\mathcal{S}_0$ and $\mathcal{S}_1$, service provider $\mathcal{P}$, and hospital $\mathcal{H}$. $\mathcal{P}$ deploys a proprietary NN model $W$, that is pre-trained on medical datasets, for medical inference. With privacy concerns, $\mathcal{P}$ uses the additive secret-sharing tool to split $W$ into two shares (i.e., $W = \langle W\rangle_0 + \langle W\rangle_1$) within the ring $\mathbb{Z}_{2^l}$, then the share $\langle W\rangle_0$ is sent to $\mathcal{S}_0$ and $\langle W\rangle_1$ is sent to $\mathcal{S}_1$. $\mathcal{H}$ holds confidential medical records $I$, and intends to leverage the well-trained NN model to facilitate a medical decision. $\mathcal{H}$ processes $I$ in a way similar to that $\mathcal{P}$ processes $W$, i.e., splitting $I$ into $\langle I\rangle_0$ for $\mathcal{S}_0$ and $\langle I\rangle_1$ for $\mathcal{S}_1$. In TrustMIS, $\mathcal{S}_0$ and $\mathcal{S}_1$ collaboratively execute secure protocols and return the shared AI-assisted inference results $\langle O\rangle$ to $\mathcal{H}$. Upon receiving the shared results $\langle O\rangle$, $\mathcal{H}$ can recover the cleartext results $O$ by calculating $O = \langle O\rangle_0 + \langle O\rangle_1$. Throughout TrustMIS, $\mathcal{H}$ learns no information except the results, while $\mathcal{P}$ learns nothing about the $\mathcal{H}$'s medical records.

To provide semantic segmentation between the background and the target, well-trained U-Nets have been prepared. There are two primary aspects for recognizing the health condition within the medical images: target data and contextual data (i.e., background). Specifically, target data encompasses the pixels within the frames that describe abnormal detection regions (such as tumors or cancers). In contrast, contextual data includes all other pixels that depict the background and visible human tissues. TrustMIS provides a binary image of the same size of the original medical image, defined as follows:

$$O(p) = \begin{cases} 0 & \text{if } p \text{ does not belong to detection regions} \\ 1 & \text{if } p \text{ does belong to detection regions} \end{cases}$$

where $p$ is the generic pixel of the medical image.

### 3.2 Threat Model and Security

TrustMIS is designed based on semi-honest (i.e., honest-but-curious) security model [5]. That is, two servers $\mathcal{S}_0$ and $\mathcal{S}_1$ strictly follow the implementation of TrustMIS's protocols but attempt to deduce more private information according to the intermediate shared results seen from each protocol execution. Like existing MPC-based PPDL works [17, 21, 25], we assume $\mathcal{S}_0$ and $\mathcal{S}_1$ are independent and non-colluding. That is, if one server is corrupted by the adversary, the other one behaves honestly.

We formally provide the security definition and proof in the real/ideal paradigm. Let $\prod$ be a protocol executed in the real interaction and $\mathcal{F}$ be the ideal functionality executed by a trusted third party. We then consider the following probabilistic experiments $\mathsf{Real}^{\prod}_{\mathcal{A}}(1^l)$ and $\mathsf{Ideal}^{\mathcal{F}}_{\mathcal{A},\mathsf{Sim}}(1^l)$, where $\mathcal{A}$ is a stateful adversary, $\mathsf{Sim}$ is a stateful simulator, and $l \in \mathbb{N}^+$ is the security parameter.

- $\mathsf{Real}^{\prod}_{\mathcal{A}}(1^l, \mathcal{S}_\theta, \langle I\rangle, \langle W\rangle)$ executes the protocol $\prod$ with security parameter $l$, where $\langle I\rangle$ and $\langle W\rangle$ are shared input from both servers and $\mathcal{S}_\theta \subset \{\mathcal{S}_0, \mathcal{S}_1\}$ is the corrupted server.
  *Output:* $(\mathsf{View}^{\prod}_{\mathcal{S}_\theta}, \langle O\rangle)$, where $\mathsf{View}^{\prod}_{\mathcal{S}_\theta}$ denotes the final view of $\mathcal{S}_\theta$ in the real world and $\langle O\rangle$ denotes the final output of $\mathcal{S}_0$ and $\mathcal{S}_1$, respectively.
- $\mathsf{Ideal}^{\mathcal{F}}_{\mathcal{A},\mathsf{Sim}}(1^l, \mathcal{S}_\theta, \langle I\rangle, \langle W\rangle)$ computes $\langle O\rangle \leftarrow \mathcal{F}(\langle I\rangle, \langle W\rangle)$.
  *Output:* $(\mathsf{Sim}^{\mathcal{F}}_{\mathcal{S}_\theta}, \langle O\rangle)$, where $\mathsf{Sim}^{\mathcal{F}}_{\mathcal{S}_\theta}$ denotes the view of the corrupted $\mathcal{S}_\theta$ generated by $\mathsf{Sim}$.

**Definition 1.** *A protocol $\prod$ semantical-securely realized $\mathcal{F}$ if for any probabilistic polynomial-time (PPT) adversary $\mathcal{A}$, there exists a PPT simulator $\mathsf{Sim}$ such that*

$$|\Pr[\mathit{Real}^{\prod}_{\mathcal{A}}(1^l) = 1] - \Pr[\mathit{Ideal}^{\mathcal{F}}_{\mathcal{A},\mathit{Sim}}(1^l)] = 1| = \mathrm{negl}(l),$$

*where* negl *denotes a negligible function.*

## 4 Approach

This section presents our proposed secure multiplication-related protocols in Section 4.1. Our protocols are employed to design secure versions of commonly-used linear layers in Section 4.2, such as convolutional and fully-connected layers. Additionally, we reformulate the transposed convolutional layer from U-Nets and then seamlessly adapt our protocols. Finally, we outline these secure layers into TrustMIS framework for MIS.

### 4.1 Supporting Protocols

In NN models, the essence of convolutional, fully-connected, and transposed convolutional layers is multiplication operations. To make TrustMIS better adapt to NNs, especially U-Nets for MIS purposes, we investigate advancements in mathematical realms and propose multiplication protocols. Specifically, we first design a secure computational-saving matrix multiplication (MM) block `SecStras` for second-power matrices with the size $2^n \times 2^n$ and $n \in \mathbb{N}^+$, as shown in **Figure 3**. There exists the trade-off between the addition/subtract operations and the $\mathsf{Mult}$ in the two-server platform. Concretely, `SecStras` block reduces the expensive multiplication operations in MM with the help of *free-computations* $+/-$ (i.e., computed locally by each server) to enhance efficiency. Toy example, for 2nd-order square matrices, `SecStras` reduces the number of element multiplications from $2\times2\times2 = 8$ to 7. Based on `SecStras` block, we then design a generic secure MM protocol `SecMaMult`, as shown in **Algorithm 1**, to reduce the number of multiplications for MM operations of arbitrary sizes. It is worth noting the underlying element-based multiplication technique employed is $\mathsf{Mult}$, which is widely utilized in various PPDL schemes [7, 15]. The details of `SecMaMult` protocol are illustrated below.

First, the "specia" greatest common factor $h$ for the dimensions $m$, $n$, and $o$ from inputs $\langle\mathbf{X}\rangle$ and $\langle\mathbf{Y}\rangle$ is learned. The term "special" signifies $h$ is constrained to be a power of 2, i.e., $h = 2^{\min(v(m),v(n),v(o))}$, where $v(x) = \max\{n \in \mathbb{N} | (2^n | x)\}$. If $h = 1$, it implies that there exists odd value(s) in $m$, $n$, and $o$. In this case,

**Algorithm 1** Secure MM protocol: `SecMaMult`.

**Input:** $\mathcal{S}_0$ and $\mathcal{S}_1$ hold shared $\langle \mathbf{X} \rangle \in \mathbb{R}^{m \times n}$ and $\langle \mathbf{Y} \rangle \in \mathbb{R}^{n \times o}$
**Output:** The shared product $\langle \mathbf{Z} \rangle = \langle \mathbf{X} \times \mathbf{Y} \rangle \in \mathbb{R}^{m \times o}$

*/* Determine the special greatest common factor */*
Initiate $h = 2^{\min(v(m), v(n), v(o))}$, where $v(x) = \max\{n \in \mathbb{N} | (2^n | x)\}$ and $\mathbb{N}$ is set of natural numbers.
**if** $h = 1$ **then**
  Pad one row/column of $\langle \mathbf{X} \rangle$ and $\langle \mathbf{Y} \rangle$ with zeros to obtain $\langle \mathbf{X}' \rangle \in \mathbb{R}^{\overline{m} \times \overline{n}}$ or/and $\langle \mathbf{Y}' \rangle \in \mathbb{R}^{\overline{n} \times \overline{o}}$, where $\overline{m}, \overline{n}, \overline{o}$ are even.
**else**
  $\langle \mathbf{X}' \rangle \in \mathbb{R}^{\overline{m} \times \overline{n}} = \langle \mathbf{X} \rangle$, $\langle \mathbf{Y}' \rangle \in \mathbb{R}^{\overline{n} \times \overline{o}} = \langle \mathbf{Y} \rangle$.
**end if**
Reset $h = 2^{\min(v(\overline{m}), v(\overline{n}), v(\overline{o}))}$.
*/* Decompose $\langle \boldsymbol{X}' \rangle$ and $\langle \boldsymbol{Y}' \rangle$ */*
Divide $\langle \mathbf{X}' \rangle$ and $\langle \mathbf{Y}' \rangle$ into $(h \times h)$-square sub-matrices $\langle \mathbf{x}' \rangle_{i,j}$ and $\langle \mathbf{y}' \rangle_{j,k}$, where $i \in [0, m']$, $j \in [0, n']$, $k \in [0, o']$, and $m' = \overline{m}/h - 1$, $n' = \overline{n}/h - 1$, $o' = \overline{o}/h - 1$.
*/* Perform computation-saving MM over sub-matrices */*
**for** $i = 1 \to [m']$ **do**
  **for** $j = 1 \to [o']$ **do**
    **for** $k = 1 \to [n']$ **do**
      $\langle \mathbf{z}' \rangle_{i,j} += \texttt{SecStras}(\langle \mathbf{y}' \rangle_{i,k}, \langle \mathbf{x}' \rangle_{k,j})$.
    **end for**
  **end for**
**end for**
$\langle \mathbf{Z}' \rangle = [\langle \mathbf{z}' \rangle_{1,1}, \cdots, \langle \mathbf{z}' \rangle_{1,o'}; \cdots; \langle \mathbf{z}' \rangle_{m',1}, \cdots, \langle \mathbf{z}' \rangle_{m',o'}]$.
$\langle \mathbf{Z} \rangle = \langle \mathbf{Z}' \rangle_{1:m,1:o}$.

we append one additional row (or column) of zeros to the matrix to ensure that the final dimensions are all even. Next, we recalculate the special greatest common factor $h$ over the modified dimensions $\overline{m}$, $\overline{n}$, and $\overline{o}$. Then, the inputs $\langle \mathbf{X}' \rangle$ and $\langle \mathbf{Y}' \rangle$ are decomposed into $(h \times h)$-square sub-matrices. $\langle \mathbf{X}' \times \mathbf{Y}' \rangle$ is now broken down into small MMs of size $h \times h$. The computation efficiency of these small MMs is improved executed by invoking the `SecStras` block. Finally, the invalid results introduced by zero-padding are discarded, resulting in the correct shared product $\langle \mathbf{Z} \rangle = \langle \mathbf{X} \times \mathbf{Y} \rangle$. By the application of the recursive method of the `SecStras` block, the computational complexity of $h \times h$ MM reduces from $\mathcal{O}(h^3)$ of standard MM to $\mathcal{O}(h^{\log_2 7}) \approx \mathcal{O}(h^{2.8})$. Finally, the computational complexity of $\mathbf{X} \times \mathbf{Y}$ becomes $\mathcal{O}(\frac{\overline{mno}}{h^3} \cdot h^{2.8}) \approx \mathcal{O}(\overline{mno}^{0.2})$, comparing to $\mathcal{O}(mno)$ traditionally.

## 4.2 Secure Layers

We now convert secure linear layers into MM to leverage `SecMaMult` for optimization.

**Secure convolutional layer** SECCONV. In order to reduce the computation of convolutional layers to an MM, the input $\langle I \rangle \in \mathbb{R}^{m \times m}$ and the multiple (i.e., $q$) filters $\langle W \rangle \in \mathbb{R}^{q \times n \times n}$ should be reshaped as follows:

$$\overrightarrow{I}_{u \cdot p + v, j \cdot n + k} = I_{j+u, k+v},$$
$$\overrightarrow{W}_{i, j \cdot n + k} = W_{i,j,k},$$

where $u \in [0, p-1]$, $v \in [0, p-1]$, $i \in [0, q-1]$, $j \in [0, n-1]$, $k \in [0, n-1]$, and $p = m - n + 1$.

Then, the convolutional operation $\mathsf{CONV}(I, W)$ can be rewritten as:

$$\mathsf{CONV}(I, W) = \overrightarrow{I} \times \overrightarrow{W}^{\top},$$

`SecStras`$(\langle \mathbf{A} \rangle, \langle \mathbf{B} \rangle)$:

**if** $|\langle \mathbf{A} \rangle| = 1 \times 1$ and $|\langle \mathbf{B} \rangle| = 1 \times 1$ **then**
  **return** $\langle \mathbf{C} \rangle = \mathsf{Mult}(\langle \mathbf{A} \rangle, \langle \mathbf{B} \rangle)$
**else**
  Decompose $\langle \mathbf{A} \rangle$ and $\langle \mathbf{B} \rangle$ into 4 equal-sized sub-matrices: $\langle \mathbf{A} \rangle = [\langle a_{0,0} \rangle, \langle a_{0,1} \rangle; \langle a_{1,0} \rangle, \langle a_{1,1} \rangle]$ and $\langle \mathbf{B} \rangle = [\langle b_{0,0} \rangle, \langle b_{0,1} \rangle; \langle b_{1,0} \rangle, \langle b_{1,1} \rangle]$.
  $\langle P_0 \rangle = \texttt{SecStras}(\langle a_{0,0} \rangle, \langle b_{0,1} \rangle - \langle b_{1,1} \rangle)$.
  $\langle P_1 \rangle = \texttt{SecStras}(\langle a_{0,0} \rangle + \langle a_{0,1} \rangle, \langle b_{1,1} \rangle)$.
  $\langle P_2 \rangle = \texttt{SecStras}(\langle a_{1,0} \rangle + \langle a_{1,1} \rangle, \langle b_{0,0} \rangle)$.
  $\langle P_3 \rangle = \texttt{SecStras}(\langle a_{1,1} \rangle, \langle b_{1,0} \rangle - \langle b_{0,0} \rangle)$.
  $\langle P_4 \rangle = \texttt{SecStras}(\langle a_{0,0} \rangle + \langle a_{1,1} \rangle, \langle b_{0,0} \rangle + \langle b_{1,1} \rangle)$.
  $\langle P_5 \rangle = \texttt{SecStras}(\langle a_{0,1} \rangle - \langle a_{1,1} \rangle, \langle b_{1,0} \rangle + \langle b_{1,1} \rangle)$.
  $\langle P_6 \rangle = \texttt{SecStras}(\langle a_{1,0} \rangle - \langle a_{0,0} \rangle, \langle b_{0,0} \rangle + \langle b_{0,1} \rangle)$.
  $\langle c_{0,0} \rangle = \langle P_4 \rangle + \langle P_3 \rangle - \langle P_1 \rangle + \langle P_5 \rangle$.
  $\langle c_{0,1} \rangle = \langle P_0 \rangle + \langle P_1 \rangle$.
  $\langle c_{1,0} \rangle = \langle P_2 \rangle + \langle P_3 \rangle$.
  $\langle c_{1,1} \rangle = \langle P_4 \rangle + \langle P_0 \rangle - \langle P_2 \rangle + \langle P_6 \rangle$.
  **return** $\langle \mathbf{C} \rangle = [\langle c_{0,0} \rangle, \langle c_{0,1} \rangle; \langle c_{1,0} \rangle, \langle c_{1,1} \rangle]$
**end if**

**Figure 3.** The proposed `SecStras` protocol.

where $\top$ denotes the transpose, and $\overrightarrow{W}^{\top}$ denotes the transpose of the matrix $\overrightarrow{W}$. In TrustMIS, all information data is over the secret-sharing domain, thus secure $\mathsf{CONV}(I, W)$ can be executed by invoking proposed secure MM protocol, i.e., $\overrightarrow{\langle O \rangle} = \texttt{SecMaMult}(\overrightarrow{\langle I \rangle}, \overrightarrow{\langle W \rangle}^{\top})$.

Finally, the output $\overrightarrow{\langle O \rangle}$ of $\texttt{SecMaMult}(\overrightarrow{\langle I \rangle}^{\top}, \overrightarrow{\langle W \rangle})$ need to be reshaped in correct shape of SECCONV outputs:

$$\text{SECCONV}(\langle I \rangle, \langle W \rangle) = \langle O \rangle_{i,j,k} \leftarrow \overrightarrow{\langle O \rangle}_{j \cdot n + k, i},$$

where $i \in [0, q-1]$, $j \in [0, p-1]$, $k \in [0, p-1]$.

**Secure transposed convolutional layer** SECTRCONV. The transposed convolution operation, often called deconvolution, is essential in U-Net models for image segmentation. It enables the decoder to upsample feature maps and produce precise segmentation results. By treating transposed convolution as an MM, we then can leverage the proposed `SecMaMult` protocol. We consider the multiple (i.e., $q$) inputs $\langle I \rangle \in \mathbb{R}^{q \times m \times m}$ and the filter $\langle W \rangle \in \mathbb{R}^{n \times n}$. The output size of $\mathsf{TrCONV}(I, W)$ is $q \times p \times p$. The derivation for the value of $p$ is as follows: $m = p - n + 1 \Rightarrow p = m + n - 1$. Initiating $\overrightarrow{W} \in \mathbb{R}^{p^2 \times m^2} = 0$ firstly, then we have:

$$\overrightarrow{I}_{i, u \cdot m + v} = I_{i,u,v},$$
$$\overrightarrow{W}_{(j+u) \cdot p + v + k, u \cdot m + v} = W_{j,k},$$

where $u \in [0, m-1]$, $v \in [0, m-1]$, $i \in [0, q-1]$, $j \in [0, n-1]$, $k \in [0, n-1]$, and $p = m + n - 1$.

Then, the transposed convolution operation $\mathsf{TrCONV}(I, W)$ can be rewritten as:

$$\mathsf{TrCONV}(I, W) = \overrightarrow{W} \times \overrightarrow{I}^{\top}.$$

In TrustMIS, the secure $\mathsf{TrCONV}(I, W)$ can be executed by invoking our proposed secure MM protocol, i.e., $\overrightarrow{\langle O \rangle} = \texttt{SecMaMult}(\overrightarrow{\langle W \rangle}, \overrightarrow{\langle I \rangle}^{\top})$.

Finally, the output $\overrightarrow{\langle O \rangle}$ of $\texttt{SecMaMult}(\overrightarrow{\langle W \rangle}, \overrightarrow{\langle I \rangle}^{\top})$ need to be reshaped into correct shape of SECTRCONV outputs:

$$\text{SECTRCONV}(\langle I \rangle, \langle W \rangle) = \langle O \rangle_{i,j,k} \leftarrow \overrightarrow{\langle O \rangle}_{j \cdot n + k, i},$$

where $i \in [0, q-1]$, $j \in [0, p-1]$, $k \in [0, p-1]$.

**Algorithm 2** TrustMIS inference framework.

**Input:** $\mathcal{S}_\theta$ holds shares $\langle I\rangle_\theta$ and $\langle W\rangle_\theta = \{\langle \grave{w}^i_u\rangle_\theta, \langle \acute{w}^i_v\rangle_\theta \mid i \in \{1,2,\cdots,d\}, u \in \{1,\cdots,u_i\}, v \in \{1,\cdots,v_i\}\}$, where $d$ denotes the U-Net's depth, $u_i$ denotes the depth of $i$-th contracting block on contracting path, $v_i$ denotes the depth of $i$-th expansive block on expansive path, $\mathcal{M}$ stores historical intermediate model parameters.

**Secure Contracting Path:**

Initiate $\langle \grave{I}^1_0\rangle = \langle I\rangle, \langle \mathcal{M}\rangle = \{\}$.
**for** $i \in \{1,\cdots,d\}$ **do**
  /* *Extracting features* */
  **for** $j \in \{1,\cdots,u_i\}$ **do**
    $\langle \grave{C}^i_j\rangle = \text{SecCONV}(\langle \grave{I}^i_{j-1}\rangle, \langle \grave{w}^i_j\rangle)$.
    $\langle \grave{I}^i_j\rangle = \text{SecRELU}(\langle \grave{C}^i_j\rangle)$.
  **end for**
  $\mathcal{S}_\theta$ computes locally $\langle \mathcal{M}\rangle_\theta = \langle \mathcal{M}\rangle_\theta \cup \langle \grave{I}^i_{u_i}\rangle_\theta$, where $\theta \in \{0,1\}$.
  /* *Down-sampling feature maps* */
  $\langle \grave{I}^{i+1}_0\rangle = \text{SecMaxPool}(\langle \grave{I}^i_{u_i}\rangle)$.
**end for**

**Secure Expansive Path:**

Initiate $\langle \acute{I}^{d-1}_0\rangle = \langle \grave{I}^d_{u_d}\rangle$.
**for** $i \in \{d-1,\cdots,1\}$ **do**
  /* *Up-sampling feature maps*/
  $\langle \acute{I}^i_0\rangle = \text{SecTrCONV}(\langle \acute{I}^i_0\rangle, \langle \acute{w}^i_0\rangle)$.
  /* *Reconstructing image* */
  **if** concatenation with feature maps from the contracting path **then**
    $\mathcal{S}_\theta$ unites $\langle \mathcal{M}_i\rangle_\theta$ and $\langle \acute{I}^i_0\rangle_\theta$ to update $\langle \acute{I}^i_0\rangle_\theta$, where $\theta \in \{0,1\}$.
  **end if**
  **for** $j \in \{1,\cdots,v_i\}$ **do**
    $\langle \acute{C}^i_j\rangle = \text{SecCONV}(\langle \acute{I}^i_{j-1}\rangle, \langle \acute{w}^i_j\rangle)$.
    $\langle \acute{I}^i_j\rangle = \text{SecRELU}(\langle \acute{C}^i_j\rangle)$.
  **end for**
  $\langle \acute{I}^{i-1}_0\rangle = \langle \acute{I}^i_{v_i}\rangle$.
**end for**
$\mathcal{S}_0$ and $\mathcal{S}_1$ jointly compute $\langle O\rangle = \text{SecCONV}(\langle \acute{I}^1_{v_1}\rangle, \langle w^*\rangle)$ to learn segmentation masks, where $w^*$ is the single filter weight of the final convolutional layer.

**Secure fully-connected layer** SecFC. The computation of fully-connected layers directly utilizes our proposed secure MM protocol with the input $\langle I\rangle \in \mathbb{R}^{m\times n}$ and the weight $\langle W\rangle \in \mathbb{R}^{n\times o}$, i.e., $\langle O\rangle = \text{SecFC}(\langle I\rangle, \langle W\rangle) = \texttt{SecMaMult}(\langle I\rangle, \langle W\rangle)$.

### 4.3 TrustMIS Framework

With the secure layers introduced above, we now outline a comprehensive scheme for trust-enhanced MIS inference. We illustrate by using the U-Net for COVID-19 segmentation in Section 6, shown in Algorithm 2[1]. Hospital $\mathcal{H}$ has original medical image $I \in \mathbb{R}$ and service provider $\mathcal{P}$ has the pre-trained U-Net model with weights $W \in \mathbb{R}$. $\mathcal{H}$ and $\mathcal{P}$ protect $I$ and $W$ as shares and send shares $\langle I\rangle_0$ and $\langle W\rangle_0$ to $\mathcal{S}_0$ and $\langle I\rangle_1$ and $\langle W\rangle_1$ to $\mathcal{S}_1$ in the ring $\mathbb{Z}_{2^l}$, respectively. Then, the dual servers $\mathcal{S}_0$ and $\mathcal{S}_1$ execute trust-enhanced MIS inference, which consists of two fundamental stages: the secure contracting path and the secure expansive path, shown in Figure 1. More specifically, the secure contracting path employs a sequence of SecCONV, SecRELU, and SecMaxPool to gradually reduce the size of feature maps while extracting high-level features from the image (lines 1 $\sim$ 11). Conversely, the expansive path restores the size of feature maps to match that of the input image through SecTrCONV, SecCONV, and SecRELU (lines 12 $\sim$ 24). Additionally, it integrates feature maps from the contracting path to enhance segmentation performance (lines 18 $\sim$ 19). The final SecCONV computes the secret-shared segmentation masks (line 25). Finally, the cleartext result $O$ of segmentation masks is obtained by merging the received shares from $\mathcal{S}_0$ and $\mathcal{S}_1$, i.e., $O = \langle O\rangle_0 + \langle O\rangle_1$.

[1] TrustMIS uses the advanced SecRELU and SecMaxPool from [19, 8] directly.

## 5 Theoretical Analysis

### 5.1 Correctness Analysis

We first provide a correctness of `SecStrass`, which is assured by Theorem 1.

**Theorem 1.** *For any input matrices $\langle \boldsymbol{A}\rangle \in \mathbb{R}^{n\times n}$ and $\langle \boldsymbol{B}\rangle \in \mathbb{R}^{n\times n}$, assume that $n = 2^k$ where $k \in \mathbb{Z}^+_0$, the* `SecStrass` *protocol yields the correct result $\langle \boldsymbol{C}\rangle$ for the matrix multiplication $\langle \boldsymbol{C}\rangle = \langle \boldsymbol{A} \times \boldsymbol{B}\rangle$.*

*Proof. 1) Correctness of the sub-protocol*: $(1\times 1)$-sized Mult($\langle \mathbf{A}\rangle$, $\langle \mathbf{B}\rangle$). Supported by precomputed multiplication triples of the form $\langle c\rangle_0 + \langle c\rangle_1 = \underbrace{a}_{\mathcal{S}_0} \cdot \underbrace{b}_{\mathcal{S}_1}$, $\mathcal{S}_0$ and $\mathcal{S}_1$ set their output shares of the multiplication as $\langle \mathbf{C}\rangle_0 = f\times a + \langle c\rangle_0$ and $\langle \mathbf{C}\rangle_1 = e\times \mathbf{B} + \langle c\rangle_1$, where the online/offline $e$ and $f$ are reconstructed by $e = \underbrace{\langle \mathbf{A}\rangle_0 - a}_{\mathcal{S}_0} + \underbrace{\langle \mathbf{A}\rangle_1}_{\mathcal{S}_1} = \mathbf{A} - a$ by $\mathcal{S}_1$ and $f = \underbrace{\langle \mathbf{B}\rangle_1 - b}_{\mathcal{S}_1} + \underbrace{\langle \mathbf{B}\rangle_0}_{\mathcal{S}_0} = \mathbf{B} - b$ by $\mathcal{S}_0$. We prove that these are the shares of $\mathbf{C}$ as follows:

$$\begin{aligned}\langle \mathbf{C}\rangle_0 + \langle \mathbf{C}\rangle_1 &= f\times a + \langle c\rangle_0 + e\times \mathbf{B} + \langle c\rangle_1\\ &= (\mathbf{B}-b)\times a + \langle c\rangle_0 + (\mathbf{A}-a)\times \mathbf{B} + \langle c\rangle_1\\ &= a\mathbf{B} - ab + \mathbf{AB} - a\mathbf{B} + \langle c\rangle_0 + \langle c\rangle_1 = \mathbf{AB}.\end{aligned}$$

*2) Inductive hypothesis*. Assume that `SecStrass` is correct for the MM of $2^n \times 2^n$ matrices.
*3) Recursive method*. let's consider the MM of $2^{n+1}\times 2^{n+1}$ matrices, and denote

$$\langle \mathbf{A}\rangle = \begin{bmatrix}\langle a_{0,0}\rangle & \langle a_{0,1}\rangle\\ \langle a_{1,0}\rangle & \langle a_{1,1}\rangle\end{bmatrix}, \langle \mathbf{B}\rangle = \begin{bmatrix}\langle b_{0,0}\rangle & \langle b_{0,1}\rangle\\ \langle b_{1,0}\rangle & \langle b_{1,1}\rangle\end{bmatrix}, \langle \mathbf{C}\rangle = \begin{bmatrix}\langle c_{0,0}\rangle & \langle c_{0,1}\rangle\\ \langle c_{1,0}\rangle & \langle c_{1,1}\rangle\end{bmatrix},$$

where $a_{i,j}$, $b_{i,j}$, and $c_{i,j}$ are $2^n \times 2^n$ sub-matrices. Then, the MM of $2^{n+1}\times 2^{n+1}$ matrices recursively decomposes into that of $2^n \times 2^n$ matrices. Taking $\langle c_{0,1}\rangle$ as an example, we have

$$\begin{aligned}\langle c_{0,1}\rangle &= \underbrace{\langle a_{0,0}\rangle \times (\langle b_{0,1}\rangle - \langle b_{1,1}\rangle)}_{\langle P_0\rangle} + \underbrace{(\langle a_{0,0}\rangle + \langle a_{0,1}\rangle)\times \langle b_{1,1}\rangle}_{\langle P_1\rangle}\\ &= \langle a_{0,0}\times b_{0,1}\rangle - \langle a_{0,0}\times b_{1,1}\rangle + \langle a_{0,0}\times b_{1,1}\rangle + \langle a_{0,1}\times b_{1,1}\rangle\\ &= \langle a_{0,0}\times b_{0,1}\rangle \in \mathbb{R}^{2^n\times 2^n} + \langle a_{0,1}\times b_{1,1}\rangle \in \mathbb{R}^{2^n\times 2^n}.\end{aligned}$$

With the *inductive hypothesis* for $2^n \times 2^n$ matrices and the correctness of $(1\times 1)$-sized Mult, we can conclude that `SecStrass` is correct for matrices of any size. □

Within the hierarchical structure of invocations, the design of `SecMaMult` invokes `SecStrass`, and the designs for linear layers (i.e., SecCONV, SecTrCONV, and SecFC) invoke `SecMaMult`. Therefore, according to the correctness of `SecStrass`, we can deduce the correctness of `SecMaMult`, SecCONV, SecTrCONV, and SecFC.

### 5.2 Security Analysis

TrustMIS's secure layers form a pipeline that incorporates a variety of cryptographic protocols for different layers, and the input and output of each layer are over the additive secret-sharing domain. TrustMIS uses Mult, SecStrass, and SecMaMult protocols for linear layers, and CryptFlow2's proven secure protocols [19] for other layers (i.e., SecRELU and SecMaxPool). We first prove the security of proposed secure linear protocols ($\prod_{\text{SecCONV}}$, $\prod_{\text{SecTrCONV}}$, and $\prod_{\text{SecFC}}$) against semi-honest adversaries under the cryptographic standard of security (**Definition 1**). Specifically, in the semi-honest model, all the values from the ideal world $\mathcal{A}$'s view (i.e., $\mathsf{Sim}_{\mathcal{A}}^{\mathcal{F}}$ generated by simulator Sim) are irrelevant to the output from the real-world protocols based on the additive secret sharing technique. And an ideal world $\mathcal{A}$'s view (i.e., $\mathsf{Sim}_{\mathcal{A}}^{\mathcal{F}}$ generated by simulator Sim) is indistinguishable from the real world adversary's view (i.e., $\mathsf{View}_{\mathcal{A}}^{\prod}$). Therefore, each proposed protocol is secure against a PPT semi-honest adversary $\mathcal{A}$. Finally, according to the *universal composability* theory [1], we can claim that the TrustMIS's inference services are secure against semi-honest adversaries, assured by Theorem 2.

**Theorem 2.** *TrustMIS's secure inference scheme $\prod^{\mathit{TrustMIS}}$ securely realizes the ideal functionality $\mathcal{F}^{\mathit{TrustMIS}}$ in the presence of one semi-honest adversary $\mathcal{A}$ in the ($\prod_{\text{SecCONV}}$, $\prod_{\text{SecTrCONV}}$, $\prod_{\text{SecFC}}$)-hybrid model.*

## 6 Experiment Evaluation

**Experimental platform and settings.** We implement a prototype of TrustMIS by using Python 3.7 and Pytorch 1.9. Extensive experiments for secure inference are conducted on two separate servers equipped with 64-core CPUs, 128GB RAM, and 2 NVIDIA GeForce RTX 2080Ti. We are modeling a local-area network environment with a network bandwidth of 1 Gbps and a network latency of 0.1 ms. We configured the integer ring size of additive secret shares as $\mathbb{Z}_{2^{32}}$ (i.e., $l = 32$) to align with baseline [21, 15, 7] settings for fairness. Additionally, we implement and train cleartext NN models using PyTorch on an NVIDIA RTX 2080Ti GPU. Each model is trained using the standard SGD optimizer with specific parameters: a learning rate of 0.001, a batch size of 128, a momentum of 0.9, and a weight decay of $1 \times 10^{-6}$.

**Datasets and models.** We evaluate TrustMIS over the commonly used MNIST and CIFAR-10 datasets, and four various medical datasets (e.g., breast cancer [2], liver disease [3], COVID-19 [4], and Brain LGG [5]). The first four datasets, MNIST, CIFAR-10, Breast cancer, and Liver disease, are utilized for disease prediction, and the last two datasets, COVID-19, and LGG, are employed for MIS. The models employed for these datasets include a 4-layer convolutional neural network (CNN) for MNIST, an 8-layer CNN for CIFAR10, a 3-layer CNN for Breast cancer and Liver disease, and 5-depth U-Nets for COVID-19 and Brain LGG. The datasets and models we used are identical to that of the baselines[15, 21, 7]. For specific details of the models, please refer to the baselines [15, 21, 7].

**Baselines and metrics.** To verify the effectiveness, we compare TrustMIS with XONN [21], MediSC [15], FastSecNet [7] for trust-aware prediction, and DP-Seg [9] for trust-enhanced segmentation, and then set micro-benchmarks for separate protocols to isolate the influence of various techniques. We assess the effectiveness of TrustMIS using the following metrics: *(1) Prediction accuracy* represents the model's accuracy in disease prediction services. *(2) Dice accuracy* represents the model's segmentation performance in MIS services. The dice accuracy is: $2\,|\mathbf{X} \cap \mathbf{Y}| \,/\, (|\mathbf{X}| + |\mathbf{Y}|)$, where $\mathbf{X}$ is the ground truth or reference segmentation, and $\mathbf{Y}$ is the segmentation produced by TrustMIS. *(3) Inference time* indicates how quickly the model can process input data and generate prediction or segmentation results. *(4) Server communication* represents the communication overhead between two servers during the process of secure inference.

[2] https://www.kaggle.com/uciml/breast-cancer-wisconsin-data.
[3] https://www.kaggle.com/uciml/indian-liver-patient-records.
[4] https://github.com/JunMa11/COVID-19-CT-Seg-Benchmark.
[5] https://www.kaggle.com/datasets/mateuszbuda/lgg-mri-segmentation?resource=download.

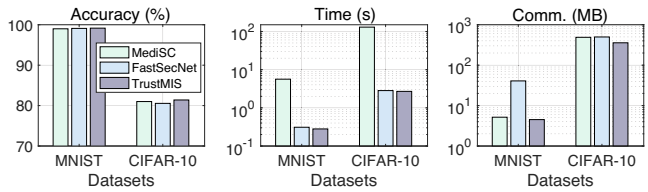


**Figure 4.** Comparison of TrustMIS's prediction with SOTA on MNIST and CIFAR-10 (Time: s, comm.: MB, acc.: %)

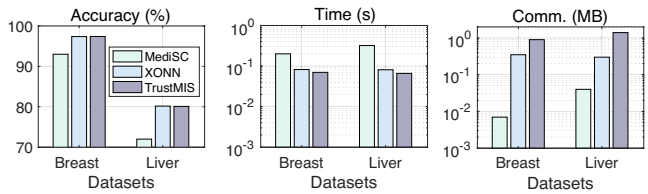


**Figure 5.** Comparison of TrustMIS's prediction with SOTA on healthcare datasets (Time: s, comm.: MB, acc.: %)

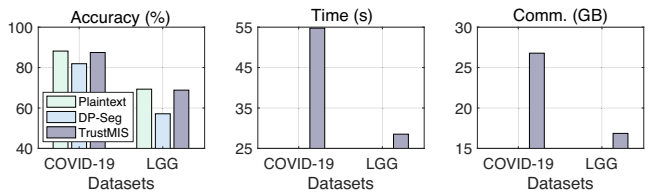


**Figure 6.** Performance summary of TrustMIS for MIS (Dice: %, time: s, comm.: GB.). The inference time from plaintext and DP-Seg schemes is similar, around $0.01 \sim 0.07$ s. Both plaintext and DP-Seg schemes incur no communication overhead since they are executed on a single server.

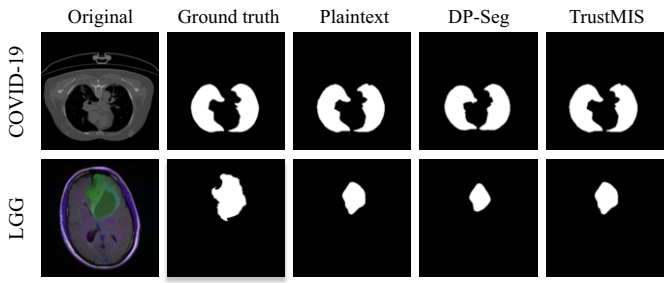


**Figure 7.** Examples of image segmentation results from various methods on healthcare datasets.

### 6.1 Evaluation of Secure Inference

**Comparison with secure prediction works**. We report the performance of TrustMIS and state-of-the-art works (SOTA) [7, 15] on benchmark datasets in Figure 4. The performance improvements of time and communication (comm.) costs are up to $1.1\times \sim 20.1\times$ and $1.1\times \sim 9.1\times$, respectively. Meanwhile, TrustMIS outperforms [7, 15] on the CIFAR-10 model (8-layer CNN), achieving improvements of up to $1.1\times \sim 54.4\times$ in time and $\sim 1.4\times$ in comm.

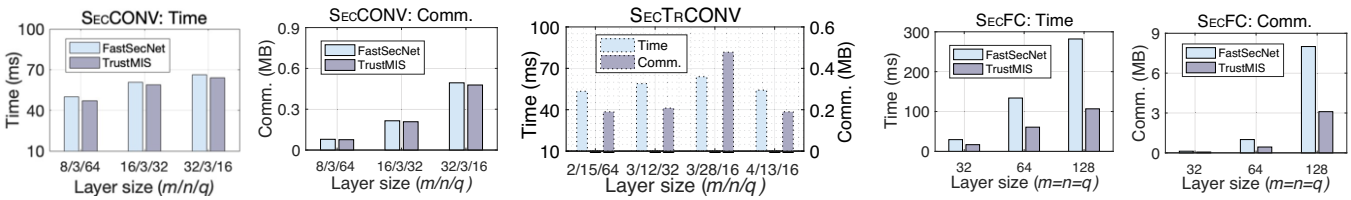


**Figure 8.** Time and comm. comparison of linear protocols with SOTA (Time: ms, comm.: MB).

The prediction performance of TrustMIS and SOTA [15, 21] on healthcare datasets is depicted in Figure 5. TrustMIS improves up to $1.2\times \sim 2.8\times$ and $1.3\times \sim 4.8\times$ speed up over breast cancer and liver disease datasets, respectively. The accuracy (acc.) performance of TrustMIS is on par with or slightly higher than that of SOTA on both benchmark and healthcare datasets.

**Performance of secure segmentation**. TrustMIS's performance for MIS is demonstrated in Figure 6. We compare TrustMIS's segmentation results (dice accuracy) with those obtained in the plaintext and DP-based [9] scenarios. For the infected lung from COVID-19 datasets, TrustMIS achieves identical segmentation results to those obtained from plaintext segmentations, i.e., 87.46% dice accuracy. TrustMIS produces $68.84\%$ segmentation accuracy on the brain LGG dataset, a slight and reasonably lower than the plaintext dice accuracy ($69.32\%$). Furthermore, compared to DP-Seg, TrustMIS significantly outperforms it, achieving an increase of $5.56\% \uparrow$ in dice accuracy on the COVID-19 dataset and $11.7\% \uparrow$ on the LGG dataset. TrustMISS's performance superiority primarily arises from its privacy protection mechanism. Unlike the DP-Seg, which relies on adding noise perturbations to protect privacy at the cost of reducing the accuracy of secure segmentation inference, TrustMISS achieves privacy preservation through a two-server model, eliminating the need to compromise between privacy and accuracy, leading to better overall performance. Each infected lung segmentation requires 21.90 s and 26.45 GB of computational and communication resources. Segmentation maps from the plaintext, DP-Seg [9], and TrustMIS are given in Figure 7.

## 6.2 *Supporting Protocols*

**Comparison with SOTA linear protocols**. The experimental results of time and comm. costs for various linear layers are listed in Figure 8. We compare the costs of SecCONV and SecFC with the counterparts implemented by the standard MM in [7, 15]. Additionally, we evaluate the time and comm. costs of SecTrCONV. However, since SecTrCONV is the first protocol tailored for secure segmentation based on the two-server model, no prior work is available for comparison in this context. SecCONV achieves $1.15\times$ improvement in both time and comm. costs. It is worth noting that SecCONV has better improvements for larger parameter values, also demonstrated in SecFC. SecFC achieves significant reductions in both time and comm. costs, saving $2\times \sim 2.64\times$ over [7] when $m = n = o$ equal to 32, 64, and 128, respectively. We demonstrate its efficiency and feasibility for SecTrCONV by showing microsecond-level time costs for various parameter configurations. For instance, when $m = 3$, $n = 28$, and $q = 16$, its runtime is 63.9 ms and the comm. cost is 0.48 MB, indicating excellent performance.

**Performance impact of activations.** We focus on 4 frequently-used activations: ReLU and its variants LeakyReLU and CeLU, and Tanh. Figure 9 evaluates the effect of activations for MIS. ReLU and its variants consistently outperform Tanh in accuracy. Among these,

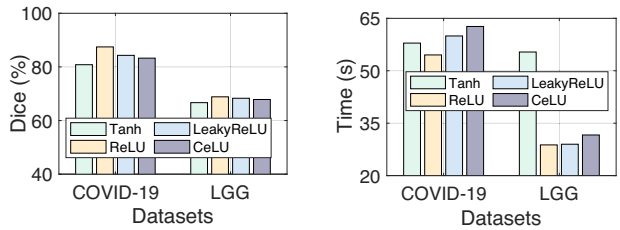


**Figure 9.** Comparison of various activations for MIS on healthcare datasets (Dice: %, Time: s)

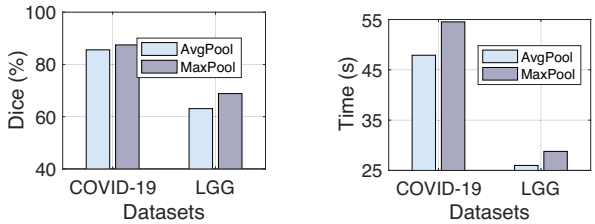


**Figure 10.** Comparison of MaxPool with AvgPool for MIS on healthcare datasets (Dice: %, time: s)

ReLU itself demonstrates slightly better dice accuracy ($0.53\% \sim 4.19\% \uparrow$) than the variants of ReLU. Our intuitive explanation for this is that activations that can eliminate negative values could be more effective for secure inference in the secret-sharing domain.

**MaxPool performs better than AvgPool**. Figure 10 reports the performance of max pooling (MaxPool) and average pooling (AvgPool) for MIS on various healthcare datasets. One U-Net model employs MaxPool for all its pooling operations, and then uses AvgPool for all its pooling layers with the same settings for comparison. We find MaxPool is better than AvgPool in TrustMIS, improving $1.88\% \sim 5.76\% \uparrow$ dice accuracy.

# 7 Conclusion

This paper presents TrustMIS, an in-the-cloud secure inference framework for MIS and prediction based on the 2PC paradigm. TrustMIS optimizes linear protocols, redesigns secure linear layers, and customizes secure transposed convolutional layers for MIS-oriented models. These designs fully rely on the lightweight additive secret-sharing tool, eliminating the need for resource-intensive cryptographic tools and enhancing performance. Security and correctness analysis of protocols in TrustMIS are proven. Our experiments on common benchmarks and real-world medical datasets demonstrate TrustMIS's practical performance. In future work, we envision extending TrustMIS to edge computing, deploying the service on user devices. Leveraging edge computing can enhance real-time processing, reduce latency, and alleviate bandwidth constraints. However, challenges such as computation resource limitations and heterogeneity in edge devices need to be addressed to ensure the scalability and efficiency of TrustMIS in edge environments.

## References


[1] R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 136–145. IEEE, 2001.

[2] K. Cheng, N. Xi, X. Liu, X. Zhu, H. Gao, Z. Zhang, and Y. Shen. Private inference for deep neural networks: A secure, adaptive, and efficient realization. *IEEE Transactions on Computers*, 2023.

[3] M. Fan, C. Chen, C. Wang, and J. Huang. On the trustworthiness landscape of state-of-the-art generative models: A comprehensive survey. *arXiv preprint arXiv:2307.16680*, 2023.

[4] M. Fan, C. Chen, C. Wang, W. Zhou, and J. Huang. On the robustness of split learning against adversarial attacks. In *ECAI 2023*, pages 668–675. IOS Press, 2023.

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

[6] G. Han, J. Choi, H. Lee, and J. Kim. Reinforcement learning-based black-box model inversion attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20504–20513, 2023.

[7] M. Hao, H. Li, H. Chen, P. Xing, and T. Zhang. FastSecNet: An efficient cryptographic framework for private neural network inference. *IEEE Transactions on Information Forensics and Security*, 18:2569–2582, 2023.

[8] Z. Huang, W. Lu, C. Hong, and J. Ding. Cheetah: Lean and fast secure two-party deep neural network inference. In *USENIX Security*, pages 809–826, 2022.

[9] M. Jiang, Y. Zhong, A. Le, X. Li, and Q. Dou. Client-level differential privacy via adaptive intermediary in federated medical imaging. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 500–510. Springer, 2023.

[10] C. Juvekar, V. Vaikuntanathan, and A. Chandrakasan. GAZELLE: A low latency framework for secure neural network inference. In *USENIX Security*, pages 1651–1669, 2018.

[11] G. A. Kaissis, M. R. Makowski, D. Rückert, and R. F. Braren. Secure, privacy-preserving and federated machine learning in medical imaging. *Nature Machine Intelligence*, 2(6):305–311, 2020.

[12] B. N. Kim, J. Dolz, P.-M. Jodoin, and C. Desrosiers. Privacy-net: an adversarial approach for identity-obfuscated segmentation of medical images. *IEEE Transactions on Medical Imaging*, 40(7):1737–1749, 2021.

[13] B. N. Kim, J. Dolz, P.-M. Jodoin, and C. Desrosiers. Mixup-privacy: A simple yet effective approach for privacy-preserving segmentation. In *International Conference on Information Processing in Medical Imaging*, pages 717–729. Springer, 2023.

[14] N. Kumar, M. Rathee, N. Chandran, D. Gupta, A. Rastogi, and R. Sharma. CrypTFlow: Secure tensorflow inference. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 336–353. IEEE, 2020.

[15] X. Liu, Y. Zheng, X. Yuan, and X. Yi. : Towards secure and lightweight deep learning as a medical diagnostic service. In *ESORICS*, pages 519–541. Springer, 2021.

[16] P. Mishra, R. Lehmkuhl, A. Srinivasan, W. Zheng, and R. A. Popa. DELPHI: A cryptographic inference service for neural networks. In *USENIX Security*, pages 2505–2522, 2020.

[17] P. Mohassel and Y. Zhang. SecureML: A system for scalable privacy-preserving machine learning. In *S&P*, pages 19–38. IEEE, 2017.

[18] K. Nandakumar, N. Ratha, S. Pankanti, and S. Halevi. Towards deep neural network training on encrypted data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 0–0, 2019.

[19] D. Rathee, M. Rathee, N. Kumar, N. Chandran, D. Gupta, A. Rastogi, and R. Sharma. CrypTFlow2: Practical 2-party secure inference. In *CCS*, pages 325–342, 2020.

[20] D. Rathee, M. Rathee, R. K. K. Goli, D. Gupta, R. Sharma, N. Chandran, and A. Rastogi. SiRnn: A math library for secure RNN inference. In *S&P*, pages 1003–1020. IEEE, 2021.

[21] M. S. Riazi, M. Samragh, H. Chen, K. Laine, K. Lauter, and F. Koushanfar. XONN:XNOR-based oblivious deep neural network inference. In *USENIX Security*, pages 1501–1518, 2019.

[22] J. Wang, D. He, A. Castiglione, B. B. Gupta, M. Karuppiah, and L. Wu. PCNNCEC: Efficient and privacy-preserving convolutional neural network inference based on cloud-edge-client collaboration. *Transactions on Network Science and Engineering*, 10(5):2906–2923, 2023.

[23] Y. Wang, Y. Luo, L. Liu, and S. Fu. pCOVID: A privacy-preserving COVID-19 inference framework. In *ICA3PP*, pages 21–42. Springer, 2023.

[24] X. Yang, J. Chen, K. He, H. Bai, C. Wu, and R. Du. Efficient privacy-preserving inference outsourcing for convolutional neural networks. *IEEE Transactions on Information Forensics and Security*, 2023.

[25] L. Zhou, Z. Wang, H. Cui, Q. Song, and Y. Yu. Bicoptor: Two-round secure three-party non-linear computation without preprocessing for privacy-preserving machine learning. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 534–551. IEEE, 2023.

[26] E. Zhu, H. Feng, L. Chen, Y. Lai, and S. Chai. Mp-net: A multi-center privacy-preserving network for medical image segmentation. *IEEE Transactions on Medical Imaging*, 2024.