Lightweight Privacy-preserving Medical Diagnosis in Edge Computing

Zhuoran Ma, Jianfeng Ma, Yinbin Miao, Ximeng Liu, *Member, IEEE*, Kim-Kwang Raymond Choo, *Senior Member, IEEE*, Ruikang Yang , and Xiangyu Wang

Abstract—With the development of machine learning, it is popular that mobile users can submit individual symptoms at any time anywhere for medical diagnosis. Edge computing is frequently adopted to reduce transmission latency for real-time diagnosis service. However, the data-driven machine learning, which requires to build a diagnosis model over vast amounts of medical data, inevitably leaks the privacy of medical data. It is necessary to provide privacy preservation. To solve above challenging issues, in this paper, we design a lightweight privacy-preserving medical diagnosis mechanism on edge, called LPME. Our LPME redesigns the extreme gradient boosting (XGBoost) model based on the edge-cloud model, which adopts encrypted model parameters instead of local data to remove amounts of ciphertext computation to plaintext computation, thus realizing lightweight privacy preservation on resource-limited edge. In addition, LPME provides secure diagnosis on edge with privacy preservation for private and timely diagnosis. Our security analysis and experimental evaluation indicates the security, effectiveness and efficiency of LPME.

Index Terms—Privacy-preserving, XGBoost, homomorphic encryption, secure computation, edge computing.

1 Introduction

Machine learning is taking an ever-increasing role in medical diagnosis, and has become prevalent for mobile users to submit symptoms at any time and then get diagnosis results. Compared with the shortage of experts and high cost in manual diagnosis, machine learning-based diagnosis has the great advantages in improving the quality of healthcare service and avoiding expensive diagnosis expenses. Thus, the construction of machine learning-based medical diagnosis has attracted much attentions from both academic and industrial fields. With the emergence of telemedicine applications, more and more demands have blossomed in healthcare [1], [2], clinical decision [3], and mobile telemedicine [4]. However, the blossom has also been accompanied by various problems, *i.e.*, the limitation of training data, vulnerabilities, and privacy concerns.

In medical practice, it is a crucial issue that the collection of enough medical data is time-consuming and expensive. A single medical origination usually stores a limited number of medical data, which is hard to support the construction of data-driven machine learning. To train an accurate diagnosis model, it is necessary to share the training data distributed among various medical institutions. With the advances of extensive storage space and unlimited computing capacity in cloud computing, machine learning over outsourced medical data has been extensively studied with the adoption of cloud [5], [6]. However, with the ever-increasing interactions between mobile users and the cloud, it incurs undesirable transmission latency and untimely request response [7], [8], [9]. A delayed diagnosis response directly influences patients' life and health as well as medical safety, especially for patients with a diagnosis for acute disease (*e.g.*, acute heart disease, pneumonia). To address this dilemma, edge computing, as a new computing paradigm, has been proposed to decrease latency and provide efficient computation service by using edge nodes [10], [11], [12], [13] which are close to mobile users. In the last few years, machine learning schemes based on edge computing [14], [15], [16] have an extensive development, which is significant to improve the diagnosis efficiency with edge computing. Fig. 1 plots a typical edge network with several edge nodes (*i.e.*, medical organizations) that owns restricted storage ability and limited computing power.

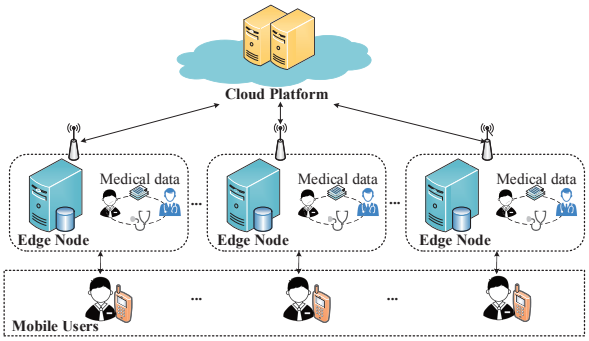


Fig. 1: The framework of edge computing for medical diagnosis.

To concentrate on the vulnerability in medical diagnosis, it is important to adopt a high-performance model on edge

- *Z. Ma, J, Ma, Y. Miao (corresponding author), R. Yang and X. Wang are with the School of Cyber Engineering, Xidian University, Xi'an 710071, China; Shaanxi Key Laboratory of Network and System Security, Xidian University, Xi'an 710071, China. E-mail: emmazhr@163.com, jfma@mail.xidian.edu.cn, ybmiao@xidian.edu.cn, ruikangY@163.com, xywang_xidian@163.com*
- *X. Liu is with the Key Laboratory of Information Security of Network Systems, College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350108, China. Email: snbnix@gmail.com*
- *K.-K. R. Choo is with the Department of Information Systems and Cyber Security, The University of Texas at San Antonio, San Antonio, TX 78249 USA. Email: raymond.choo@fulbrightmail.org*

for real-time and reliable medical diagnosis. Extreme gradient boosting (XGBoost) as the most state-of-the-art machine learning model enjoys the excellent prediction performance in the distributed setting, which demonstrates the outstanding ability in Kaggle competitions. Besides, with the tree-based structure, XGBoost has the advances of explainability and ease of understanding. Therefore, there are a large number of schemes applied the XGBoost model for medical diagnoses [17], [18], [19], but they ignore the important issue of data privacy during the training phase. Actually, patients diagnosed with private diseases (*e.g.,* HIV, Hepatitis B virus) usually bear some psychological barrier when the diagnosis results are leakage to others. It is considered as a cause to worsen the condition. Thus, it is necessary to provide privacy preservation for them. Besides, the medical data contain a large amount of sensitive information, with the release of privacy policies (*i.e.,* GDPR [20] and HIPPA [21]), more and more data are forbidden to transform in the form of plaintext. Therefore, it is urgent to protect privacy of medical diagnosis in the edge computing environment [22], [23], [24], [25].

To address privacy concerns, Homomorphic encryption (HE) [26] is a promising solution to avoid privacy leakage risks while protecting data confidentiality. The existing privacy-preserving machine learning mechanisms based on HE mainly rely on the cloud computing framework with single-cloud model or dual-cloud mode, which has been extended to edge computing [27]. Unfortunately, the single-cloud model [28] is more vulnerable to lead the privacy leakage compared with dual-cloud model, as the secret key is stored in the single cloud. Once the cloud is compromised, the sensitive information are disclosed. Besides, the strong assumption of non-collusion between two semi-honest cloud servers defined in the dual-cloud model limits its practical applications [29], [30], [31]. In addition, the training phase of machine learning involves amounts of secure computation over encrypted data. With the increasing outsourced-encrypted data, it bears a heavy computation burden [32], [33], especially for resource-constrained edge nodes, which is the first challenging issue. Therefore, it is critical to take lightweight into consideration with privacy-preserving machine learning in edge computing.

To address the above challenges, we design a lightweight privacy-preserving XGBoost over encrypted model parameters to greatly lighten computational overhead, compared with data sharing-based privacy-preserving machine learning. In this paper, we present the **L**ightweight **P**rivacy-preserving **M**edical diagnosis in **E**dge computing, which is termed as LPME. Specifically, our LPME mainly has the following constructions:

- *Lightweight XGBoost on edge*: LPME system constructs a XGBoost-based diagnosis model with model parameters trained over multiple edge nodes rather than training data, which not only eliminates the drawbacks of burdensome training data storage, but also guarantees the feasibility of XGBoost.
- *Privacy-preserving training*: LPME system designs HE-based secure computation with a single-cloud model, which selects optimal parameters over encrypted model parameters during the training phase. Since the secret key is randomly split into two parts, only one is stored in the single cloud. Thus, the single-cloud model can not only provide strong privacy preservation for training the lightweight XGBoost, but also guarantee the reliability of the privacy-preserving training on the resource-limited edges.
- *Secure diagnosis on XGBoost at edge*: LPME system provides secure diagnosis, in which a mobile user can submit his/her encrypted requests to an edge, then the edge will return the corresponding diagnosis results. During the process, HE is adopted to guarantee confidentiality of the returned diagnosis results for implementing the private and timely diagnosis.

2 Related Work

Earlier work on privacy-preserving machine learning [34], [35] has been proposed to provide privacy preservation during the training phase, but these schemes lacked implementation. Thereafter, increasing schemes have been proposed to provide privacy protection. Fu *et al.* [36] presented a privacy-preserving non-negative matrix factorization method based on addition HE, which supports matrix factorization with encrypted data, but these matrix parameters can be obtained by another party during the computation process, it will lead to the potential privacy leakage. Ma *et al.* [29] proposed a privacy-preserving random tree framework with Paillier cryptosystem, which implemented accurate and secure training over encrypted data. Wang *et al.* [32] presented a privacy-preserving collaborative neural network scheme to construct a model without privacy disclose. Mohassel *et al.* [33] designed a privacy-preserving system for efficient training neural networks. The above schemes [29], [32], [33] all adopt the HE-based mechanism, which are feasible for machine learning with privacy preservation. However, the secure computation implemented over large number of encrypted data leads to high computation overhead [31].

To solve the above issue, model sharing-based privacy-preserving machine learning framework has been designed, which outsources encrypted model parameters rather than a larger number of local data. It can not only guarantee the training of machine learning, but also move amounts of outsourced computation over ciphertexts to local computation over plaintexts, which can significantly improve the efficiency and reduce the computation burden. Yu *et al.* [37] first introduced a framework based on outsourced models from multiple data owners without disclosing local data. However, this framework uses random numbers rather than encryption technology, which is highly susceptible to inference attack resulting in privacy leakage [38]. After that, Cheng *et al.* [39] proposed a secure XGBoost over encrypted model parameters. However, these parameters can be decrypted and obtained by another party. Due to parameters also contain sensitive information, it can threaten the security of local data. Li *et al.* [40] proposed a secure classification service with the outsourced encrypted Support Vector Machine (SVM) models, but it cannot implement privacy-preserving model training. Aono *et al.* [41] presented a privacy-preserving deep learning system based encrypted models without revealing the participants' local

data to a server, which greatly reduced execution times of involved secure computation with remaining the accuracy. Unfortunately, Wang *et al.* [42] demonstrated that the aforementioned schemes [40], [41] will bring a privacy leakage in the single-cloud model. Due to the privacy of trained model, it is easy to leak when the cloud is compromised.

To avoid the limitation in the single-cloud model, the dual-cloud model is employed to prevent the computing process from privacy leakage. Liu *et al.* [31], [30] demonstrated the security and accuracy of secure computation with dual-cloud server model. Besides, Hu *et al.* [28] showed that the non-colluding dual-cloud model achieved a higher level of security compared with the single-cloud model. Even one server is compromised, it still cannot leak the privacy of trained model with the existence of the other server. With the privacy requirements in edge computing, Liu *et al.* [27] extended the secure computation based on a dual-cloud model to the environment of edge computing. Unfortunately, encrypted data should be transmitted between two cloud servers for secure computation, which will incur the communication burden and heavy computational overhead. Besides, each resource-limited edge node involves five modular exponentiation operations, two modular addition operations and six modular multiplication operations for secure computation, which is impractical in the edge computing environment. Zhang *et al.* [43] proposed a privacy-preserving feature transform on edge with lightweight, but the submitted images were presented as the form of plaintext, which cannot guarantee strong privacy preservations. To the best of our knowledge, existing literatures do not take the tradeoff between privacy concerns and lightweight into consideration in the edge computing [22], [44], [45]. Apart from achieving efficiency and real-time model training, we also devise a lightweight privacy-preserving machine learning scheme with strong privacy preservations on edge.

TABLE 1: Functionalities, securities and techniques in various schemes: A comparative summary

Functions	Fun_1	Fun_2	Fun_3	Fun_4	Fun_5	Fun_4
[29]	Data	Random forest	Dual-cloud	✓	✗	✗
[32]	Data	Neural network	Dual-cloud	✓	✗	✗
[33]	Data	Neural network	Dual-cloud	✓	✗	✗
[40]	Model	SVM	Single-cloud	✗	✗	✗
[41]	Model	Deep learning	Single-cloud	✗	✓	✗
[27]	Data	Deep learning	Dual-cloud	✓	✗	✓
LPME	Model	XGBoost	Single-cloud	✓	✓	✓

Notes. Fun_1: Data or Model sharing-based privacy preservation; Fun_2: Machine learning algorithm; Fun_3: Single-cloud or Dual-cloud secure computation model; Fun_4: Whether supporting strong security or not; Fun_5: Whether achieving lightweight transmission or not; Fun_6: Whether supporting edge computing or not.

TABLE 1 summarizes the comparison between LPME system and previous privacy-preserving machine learning schemes [29], [32], [33], [40], [41], [27]. It reveals that LPME provides not only a dual-server model of strong secure computation based on edge, but also lightweight privacy-preserving machine learning based on secure model sharing.

3 PRELIMINARY

This section describes the XGBoost [46], [39] as the basic of machine learning algorithm, and then defines two-trapdoor public-key cryptosystem [30] as the basic cryptosystem in LPME system.

3.1 XGBoost Algorithm

Given a training dataset $X \in \mathbb{R}^{n\times d}$ with n samples and d features, the object function of XGBoost in t-th round is represented as

$$Obj^{(t)} \simeq \sum_{i=1}^{n}\{\ell(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2}h_i f_t^2(x_i)\} + \Omega(f_t),$$
$$g_i = \partial_{\hat{y}^{(t-1)}}\ell(y_i, \hat{y}_i^{(t-1)}), h_i = \partial^2_{\hat{y}^{(t-1)}}\ell(y_i, \hat{y}_i^{(t-1)}),$$

where g_i represents the first gradient of loss function ℓ, and h_i represents the second gradient of ℓ. The regularization $\Omega(f_t) = \gamma T + \frac{1}{2}\psi\|w\|^2$ measures the complexity of model, where T means the number of leaf nodes.

Besides, the logistic loss ℓ of training loss measures how well model fits on training data, as demonstrated in

$$\ell(y_i, \hat{y}_i^{(t-1)}) = y_i \ln(1 + e^{-\hat{y}_i}) + (1 - y_i)\ln(1 + e^{\hat{y}_i}).$$

Assume that a XGBoost model XGB contains K trees. Given the $i-$th training sample $x_i \in \mathbb{R}^d$, the corresponding prediction $\hat{y}_i$ is computed as

$$\hat{y}_i = \sum_{k=1}^{K}\mathcal{F}_k(x_i),$$
$$s.t.\, \mathcal{F}_k \in \mathrm{XGB}, \text{ where } \mathrm{XGB} = \{\mathcal{F}_1, \mathcal{F}_2, ..., \mathcal{F}_K\}.$$

3.2 Two-trapdoor Cryptosystem

The two-trapdoor public-key cryptosystem contains five algorithms as follows.

- $(pk, sk) \leftarrow \mathtt{KeyGen}(\Im)$: Given the security parameter $\Im$, distinct odd primes p, q are generated, where $|p| = |q| = \Im$, $\eta = pq$, and $\lambda = lcm(p-1, q-1)$. The public key $pk = (\eta, \varrho)$ and secret key $sk = \lambda$ are generated, where $\varrho = (1+\eta) \mod \eta^2$.
- $sk^{(1)}, sk^{(2)} \leftarrow \mathtt{KeyS}(sk)$: The secret key $sk = \lambda$ is randomly divided into two secret shares $sk^{(1)}$ and $sk^{(2)}$ satisfying $\sum_{i=1}^{2} sk^{(i)} \equiv 0 \mod \lambda$ and $\sum_{i=1}^{2} sk^{(i)} \equiv 1 \mod \eta^2$.
- $[\![x]\!] \leftarrow \mathtt{Enc}_{pk}(x)$: Given a plaintext x, it outputs an encrypted data $[\![x]\!] = \varrho^x \cdot r^\eta \mod \eta^2$ with a public key pk, where $r \in \mathbb{Z}^*_{\eta^2}$ denotes a random number.
- $x \leftarrow \mathtt{Dec}_{sk}([\![x]\!])$: Given an encrypted data $[\![x]\!]$, it decrypts the corresponding plaintext x with secret key sk, where $x = \frac{[\![x]\!]^\lambda \mod \eta^2 - 1}{\eta}\lambda^{-1} \mod \eta$.
- $[\![x]\!]^{(i)} \leftarrow \mathtt{SDec}_{sk^{(i)}}([\![x]\!])$: Given an encrypted data $[\![x]\!]$, it outputs the corresponding decryption share $[\![x]\!]^{(i)}$ with a secret share $sk^{(i)}$, where $[\![x]\!]^{(i)} = [\![x]\!]^{sk^{(i)}} \mod \eta^2$.
- $x \leftarrow \mathtt{WDec}(\{[\![x]\!]^{(1)}, [\![x]\!]^{(2)}\})$: Given the tuple of decryption shares $\{[\![x]\!]^{(1)}, [\![x]\!]^{(2)}\}$, it outputs the decryption x, where $x = \frac{\prod_{i=1}^{2}[\![x]\!]^{(i)} \mod \eta^2 - 1}{\eta}$.

Besides, the cryptosystem based on addition homomorphic owns two properties as follows:

$$[\![x_1 + x_2]\!] = [\![x_1]\!] \cdot [\![x_2]\!] \text{ and } [\![-x]\!] = [\![x]\!]^{\eta-1}. \quad (1)$$

3.3 Secure Computation

Here, we introduce the Secure Multiplication (SMUL) and Secure Comparison (SCOM) operations for secure computation. Suppose that there are two semi-honest parties (*i.e.*, Alice and Bob) in the multiplication and comparison over encrypted data, the goals of SMUL and SCOM are that all intermediate results and final computation results cannot be disclosed to both parties. Given two encrypted numbers $[\![x_1]\!]$ and $[\![x_2]\!]$, Alice holds a secret share $sk^{(1)}$, Bob holds the other secret share $sk^{(2)}$. SMUL and SCOM are defined as follows:

SMUL$([\![x_1]\!], [\![x_2]\!]) \rightarrow [\![x_1 \times x_2]\!]$: Alice first generates $[\![x_1']\!] = [\![x_1]\!] \cdot [\![r_1]\!]$, $[\![x_2']\!] = [\![x_2]\!] \cdot [\![r_2]\!]$, where $r_1, r_2 \in \mathbb{Z}_{\eta^2}^*$ are two random numbers, then uses $\mathtt{SDec}_{sk^{(1)}}$ to obtain $[\![x_1']\!]^{(1)}$ and $[\![x_2']\!]^{(1)}$. On receiving these encrypted data, Bob uses SDec and WDec with $sk^{(2)}$ to obtain x_1' and x_2', and computes $[\![res]\!] = x_1' \times x_2'$. Then, Alice runs $[\![x_1 \times x_2]\!] = [\![res]\!] \cdot [\![r_1 \times r_2]\!]^{\eta-1} \cdot [\![x_1]\!]^{\eta-r_2} \cdot [\![x_2]\!]^{\eta-r_1}$ to remove random numbers, and the multiplication result $[\![x_1 \times x_2]\!]$ is returned.

SCOM$([\![x_1]\!], [\![x_2]\!]) \rightarrow res$: Alice first calculates $[\![x_1']\!] = [\![x_1]\!]^2 \cdot [\![1]\!]$, $[\![x_2']\!] = [\![x_2']\!]^2 \cdot [\![1]\!]$, and runs $[\![res]\!] \leftarrow ([\![x_1']\!] \cdot [\![x_2']\!]^{\eta-1})^{r_1} \cdot [\![r_2]\!]$, where $r_1, r_2 \leftarrow \mathbb{Z}_\eta$ ($r_2 \ll r_1$) are two random numbers. Then, $[\![res]\!]^{(2)} \leftarrow \mathtt{SDec}_{sk^{(1)}}([\![res]\!])$ is obtained. After involving the SDec and WDec algorithms, Bob obtains res via computing the bit length of res as Eq. 3, and returns the comparison result.

$$res = \begin{cases} x_1 < x_2, & |res| > |\eta|/2; \\ x_1 \geq x_2, & otherwise. \end{cases} \tag{2}$$

4 Problem Formulation

In this section, we define system model, threat model and design goals of LPME system, respectively.

4.1 System Model

Our system model mainly involves four entities: Key Generation Center (KGC), Cloud Platform (CP), Edge Nodes (ENs), and Mobile Users (MUs), which is demonstrated in Fig. 2. Assume that N ENs are contained in the system. Note that the communication among these entities is synchronized with a secure channel, such as Secure Socket Layer (SSL) and Transport Layer Security (TLS). The concrete role of each entity is shown as follows:

- *Key generation center*. KGC is fully trusted to generate, manage, and distribute secret keys for our system, where the secret shares are sent to other entities for future secure computation (Step ①).
- *Edge node*. An EN, which stores limited medical data, is a medical institution with the constraints of storage space and computation capacity. During the training phase, an EN is willing to collaboratively build a global model with other ENs, which submits locally optimal model parameters after encryption, and provides computation service for CP to implement secure computation (Step ②).
- *Cloud platform*. CP has unlimited computation and storage capacities. It first receives the encrypted model parameters from multiple ENs, and then chooses the globally optimal model parameters for the construction of a global model (Step ③).
- *Mobile user*. A MU is able to submit an encrypted request to a nearby EN for diagnosis (Step ④), and obtain the encrypted diagnosis result returned by the EN (Step ⑤). To protect diagnosis privacy, secure computation of the diagnosis phase is implemented between a MU and the EN together.

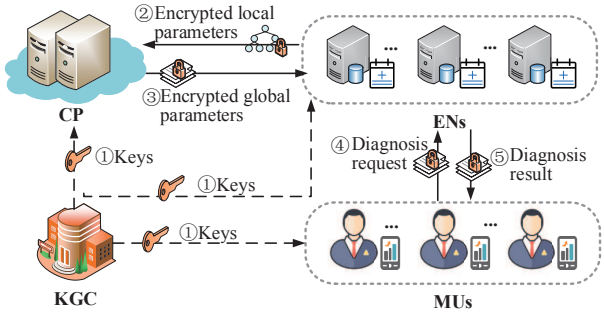


Fig. 2: System model.

4.2 Threat Model

For the adversarial perspective, we consider the possible threats of the system depending on the information accessed by entities (*i.e.*, CP, ENs, and MUs) in the system.

- *Threats from entities*: Assume that KGC is trustable for key distribution. MUs, ENs, and CP are considered as *honset-but-curious* entities that honestly follow specified protocols but attempt to obtain additional information from encrypted data. In practice, the collusion between CP and ENs reveals the privacy of ENs while the collusion between an EN and a MU also discloses individual sensitive information. Therefore, CP, EN, and MU are not worth colluding with other entities to avoid individual privacy leakage. We assume that there is no collusion between CP and ENs, ENs and MUs, respectively.
- *Threats from external adversary*: We assume that the external adversary has the ability to eavesdrop on the transmitted information from the communication channels between CP and ENs, MUs and ENs, respectively. And also an adversary can corrupt an EN or a MU or the CP.

4.3 Design Goals

Our system aims to achieve a privacy-preserving machine learning framework with secure training, accurate diagnosis, and lightweight computation under the adversarial environment. Design goals are shown in following:

- *Security*. (a) Each locally-built model from an EN contains sensitive information that cannot be disclosed for model privacy. (b) The parameters and intermediate computation results cannot be leaked during the construction process of the global model. (c) All MU-submitted requests sent to ENs and the

corresponding diagnosis results are only known to the MU for diagnosis privacy.

- *Efficiency*. LPME system should guarantee the efficiency of medical diagnosis with the trained global model and keep the lightweight workload on the ENs and MUs.
- *Effectiveness*. LPME system should keep reliable and accurate diagnosis service, which is of great significance to provide accurate diagnosis results for MUs.

5 LPME Framework

In this section, we detailedly describe how to construct a basic lightweight privacy-preserving XGBoost framework for the global diagnosis model, then design secure diagnosis on edge to provide the private and timely diagnosis service. The notation definitions are described in Table 2.

TABLE 2: Notation descriptions in LPME system

Notations	Descriptions
$sk^{(1)}, sk^{(2)}$	Secret shares for CP and ENs
λ_1, λ_2	Secret shares for ENs and MUs
η	Security parameters
$[\![x]\!]^{(1)}, [\![x]\!]^{(2)}$	Decryption shares for $[\![x]\!]$
XGB	Trained XGBoost model
$\hbar$	Tree height
d	Feature dimension of training data
$\uparrow$	Numerator symbol
$\downarrow$	Denominator symbol
f^*, s^*	Split threshold with the value s^* on f^*-th feature
KeyGen,KeyS	Key generation and key split algorithms
Enc,Dec	Encryption and decryption algorithms
SDec,WDec	Partial decryption and full decryption algorithms
SCOM	Secure comparison over two encrypted data
SMUL	Secure multiplication over two encrypted data
cons	Approximate precision

5.1 Overview

Fig. 3 illustrates the proposed process of LPME system. The process consists of three principal stages:

- *Key generation*: To provide privacy preservations, KGC first employs KeyGen to generate a key pair (pk, sk). Then, KeyS(sk) is invoked twice to randomly split sk into secret shares, *i.e.*, $(sk^{(1)}, sk^{(2)}) \leftarrow$ KeyS(sk) and $(\lambda_1, \lambda_2) \leftarrow$ KeyS(sk), where $(sk^{(1)}, sk^{(2)})$ satisfy $sk^{(1)} + sk^{(2)} \equiv 0 \mod \lambda$ and $sk^{(1)} + sk^{(2)} \equiv 1 \mod \eta^2$, (λ_1, λ_2) satisfy $\lambda_1 + \lambda_2 \equiv 0 \mod \lambda$ and $\lambda_1 + \lambda_2 \equiv 1 \mod \eta^2$. In addition, KGC distributes $\{sk^{(1)}, pk\}$ to the CP, $\{sk^{(2)}, \lambda_1, pk\}$ to ENs, and $\{\lambda_2, pk\}$ to MUs.
- *Lightweight privacy-preserving XGBoost*: To construct a global model over N ENs, ENs first locally train and encrypt decision nodes before sending them to CP (Step ①). Then, CP chooses the best split of a decision node among submitted-nodes as the global node to achieve global optimization (Step ②). Finally, each EN builds local leaf nodes (Step ③).
- *Secure diagnosis on edge*: To implement a secure diagnosis service, a MU is required to encrypt symptoms before transmitting them to a nearby EN. It is necessary to protect the confidentiality of the submitted-symptoms and the returned diagnosis results.

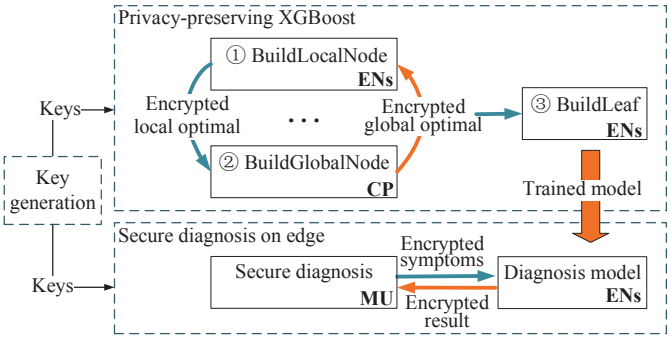


Fig. 3: The overview of LPME

5.2 Lightweight Privacy-preserving XGBoost

We assume that multiple ENs collaboratively construct the global model without sharing training data. Without loss of generality, the data stored in multiple ENs are considered as non-i.i.d distribution, *i.e.*, they have the global distribution but also maintain individual biased distribution. Therefore, the final trained model of each EN not only learns knowledge over all ENs, but also remains local differences. Specifically, the proposed privacy-preserving XGBoost is built over N ENs. During the training of k-th round, the k-th tree model is represented as $\mathcal{F}_k(x) = w_q(x)$, where tree nodes are divided into decision nodes and leaf nodes, and each decision node contains a split value. Fig. 4 shows the concrete process of building a tree with the tree height $\hbar = 3$ It is necessary to build a tree from a root node down to leaf nodes.

5.2.1 BuildDecisionNode

To construct decision nodes over N ENs, the specific process is divided into **BuildLocalNode** (*i.e.*, building decision nodes on a local EN) and **BuildGlobalNode** (*i.e.*, building decision nodes with ENs on CP).

BuildLocalNode: For the i-th ($i \in [1, N]$) EN, the best splitting is selected with maximization of $Gain$. As shown in

$$Gain = \frac{1}{2} \times gain - \psi,$$
$$gain = \frac{G_L^2}{H_L + \psi} + \frac{G_R^2}{H_R + \psi} - \frac{(G_L + G_R)^2}{H_L + H_R + \psi},$$
$$G_L = \sum_{i \in X_L} g_i,\ G_R = \sum_{i \in X_R} g_i,$$
$$H_L = \sum_{i \in X_L} h_i,\ H_R = \sum_{i \in X_R} h_i,$$

X_L and X_H denote the sample space of the leaf and right tree nodes after the splitting, respectively.

Suppose that a set of split candidates is represented as $\{S_f\}_{f=1}^d$, the split threshold s_m^* on the f_m^*-th feature with the maximal $Gain$ is locally optimal split. The specific process is shown in **Algorithm** 1. For the construction of the globally optimal split, the i-th EN sends locally optimal split $[\![s_i^*]\!]$, $[\![f_i^*]\!]$ and the evaluation index $[\![gain_i]\!] = ([\![\alpha_i^{\uparrow}]\!], [\![\alpha_i^{\downarrow}]\!])$ to CP after encryption, where $gain_i = \frac{\alpha_i^{\uparrow}}{\alpha_i^{\downarrow}}$ and $\alpha_i^{\downarrow} > 0$.

For example, a local node is trained over the i-th EN to yield $s_i^* = 1.25$, $f_i^* = 1$ and $gain_i = -22.5$. Due to

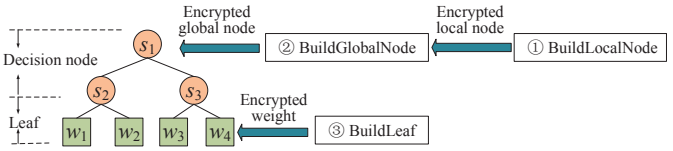


Fig. 4: The concrete process of building a tree.

the encryption needs to be implemented over integers, the i-th EN approximates rational numbers to integers. $cons$ denotes the precision of approximation. *i.e.,* $[\![s_i^*]\!] = [\![125]\!] \leftarrow \mathtt{Enc}_{pk}(1.25 * cons)$[1], and $gain_i = -\frac{45}{2}$, where $\alpha_i^{\uparrow} = 45$ and $\alpha_i^{\downarrow} = 2$ before encryption.

Algorithm 1: Locally Optimal Split

Input: Sample space X with d features on current node.
Output: Optimal local split $[\![f^*]\!]$,$[\![s^*]\!]$, with $[\![\alpha^{\uparrow}]\!]$ and $[\![\alpha^{\downarrow}]\!]$.

$score \leftarrow 0$;
$f^*, s^* \leftarrow 0$;
$\alpha^{\uparrow} \leftarrow 0, \alpha^{\downarrow} \leftarrow 1$;
for $0 < f \leq d$ **do**
  for $s_j \in S_f$ **do**
    /* S_f is the set of split candidate on f-th feature */
    if $score < Gain$ **then**
      $score \leftarrow Gain$;
      /* $gain = \frac{\alpha^{\uparrow}}{\alpha^{\downarrow}}$ */
      $\alpha^{\uparrow} \leftarrow G_L^2(H_R + \psi) + G_R^2(H_L + \psi)(H_L + H_R + \psi) - (G_L + G_R)^2(H_L + \psi)(H_R + \psi)$;
      $\alpha^{\downarrow} \leftarrow (H_L + \psi)(H_R + \psi)(H_L + H_R + \psi)$;
      $f^* \leftarrow f$;
      $s^* \leftarrow s_j$;
$\mathtt{Enc}_{pk}(\cdot)$ is called to encrypt $f^*, s^*, \alpha^{\uparrow}, \alpha^{\downarrow}$;
return $[\![f^*]\!], [\![s^*]\!], [\![\alpha^{\uparrow}]\!], [\![\alpha^{\downarrow}]\!]$.

Before introducing **BuildGlobalNode**, we design **CompareEncIndex** to compare encrypted evaluation indexes $[\![gain_i]\!]$ and $[\![gain_j]\!]$ of two decision nodes to find the decision node with the maximum $gain$ for global optimization. Two decision nodes are computed with **BuildLocalNode** by two ENs (*i.e.,* denoted as EN_i and EN_j).

CompareEncIndex: To compare two values of $[\![gain_i]\!]$ and $[\![gain_j]\!]$, we illustrate the specific process as

$$\frac{\alpha_i^{\uparrow}}{\alpha_i^{\downarrow}} - \frac{\alpha_j^{\uparrow}}{\alpha_j^{\downarrow}} = \frac{\alpha_i^{\uparrow} \times \alpha_j^{\downarrow} - \alpha_i^{\downarrow} \times \alpha_j^{\uparrow}}{\alpha_i^{\downarrow} \times \alpha_j^{\downarrow}} \quad (\alpha_i^{\downarrow} \times \alpha_j^{\downarrow} > 0).$$

For simplification, if $\alpha_i^{\uparrow} \times \alpha_j^{\downarrow} - \alpha_i^{\downarrow} \times \alpha_j^{\uparrow} < 0$, then $gain_i < gain_j$; Otherwise, $gain_i \geq gain_j$. The comparison process is executed by computing $[\![\alpha_i^{\uparrow} \times gain_j^{\downarrow}]\!]$, $[\![\alpha_i^{\downarrow} \times gain_j^{\uparrow}]\!]$, $[\![val]\!] = [\![\alpha_i^{\uparrow} \times \alpha_j^{\downarrow}]\!] \cdot [\![\alpha_i^{\downarrow} \times \alpha_j^{\uparrow}]\!]^{\eta-1}$. Then, $res \leftarrow \mathtt{SCOM}([\![val]\!], [\![0]\!])$ is obtained.

If $res = 0$, then $gain_i \geq gain_j$; Otherwise $gain_i < gain_j$. The whole process involves secure multiplication (SMUL) and secure comparison computation (SCOM) between CP and ENs, where CP holds $sk^{(1)}$ and ENs hold $sk^{(2)}$. The specific process is shown as follows.

1. Note that $cons = 100$.

Step 1. CP first randomly chooses two numbers $r_1, r_2 \in \mathbb{Z}_{\eta}$, and then computes the blinded numbers as $[\![\alpha_i^{\uparrow\prime}]\!] \leftarrow [\![\alpha_i^{\uparrow}]\!] \cdot [\![r_1]\!], [\![\alpha_i^{\downarrow\prime}]\!] \leftarrow [\![\alpha_i^{\downarrow}]\!] \cdot [\![r_1]\!]$.

Similarly, $[\![\alpha_j^{\uparrow\prime}]\!]$ and $[\![\alpha_j^{\downarrow\prime}]\!]$ are computed by blinding r_2. Besides, decryption shares are computed with $\mathtt{SDec}_{sk^{(1)}}(\cdot)$, then $[\![\alpha_i^{\uparrow\prime}]\!]$, $[\![\alpha_j^{\downarrow\prime}]\!]$ and the corresponding decryption shares are sent to EN_i, while $[\![\alpha_j^{\uparrow\prime}]\!]$, $[\![\alpha_i^{\downarrow\prime}]\!]$ and the corresponding decryption shares are sent to EN_j.

Step 2. Once receiving the encrypted data, each EN operates with $\mathtt{SDec}_{sk^{(2)}}(\cdot)$ and $\mathtt{WDec}(\cdot)$ to obtain the blinded numbers, and then returns the multiplication result $[\![res']\!]_{pk}$ over the blinded numbers to CP.

Step 3. CP removes the blinded random numbers r_1, r_2 to obtain the encrypted multiplication results $[\![\alpha_i^{\uparrow} \times \alpha_j^{\downarrow}]\!]$ and $[\![\alpha_i^{\downarrow} \times \alpha_j^{\uparrow}]\!]$. More details of SMUL are shown in [30]. After that, CP obtains $[\![val]\!]$ computed as

$$[\![val]\!] = [\![\alpha_i^{\uparrow} \times \alpha_j^{\downarrow} - \alpha_i^{\downarrow} \times \alpha_j^{\uparrow}]\!], = [\![\alpha_i^{\uparrow} \times \alpha_j^{\downarrow}]\!] \cdot [\![\alpha_i^{\downarrow} \times \alpha_j^{\uparrow}]\!]^{\eta-1},$$

then calculates $[\![2 \cdot val + 1]\!] = [\![val]\!]^2 \cdot [\![1]\!]$.

Besides, two numbers $r_1', r_2' \in \mathbb{Z}_{\eta}$ ($r_2' \ll r_1'$) are randomly selected, then $[\![2r_1' \cdot val + r_2']\!] \leftarrow ([\![2val+1]\!] \cdot [\![1]\!]^{\eta-1})^{r_1'} \cdot [\![r_2']\!]$.

Finally, CP obtains $[\![2r_1' \cdot val + r_2']\!]^{(1)} \leftarrow \mathtt{SDec}_{sk^{(1)}}([\![2r_1' \cdot val + r_2']\!])$ before sending $[\![2r_1' \cdot val + r_2']\!]$ and its decryption shares to an EN (*i.e.,* EN_i or EN_j, according to the idle state).

Step 4: With SDec and WDec algorithms, the EN obtains $2r_1' \cdot val + r_2'$. After computing as

$$res = \begin{cases} 1, & |2r_1' \cdot val + r_2'| > |\eta|/2 \\ 0, & otherwise \end{cases}, \tag{3}$$

the EN encrypts res and sends $[\![res]\!]$, $[\![res]\!]^{(1)} \leftarrow \mathtt{SDec}_{sk^{(1)}}([\![res]\!])$ to CP. Then, CP obtains the final comparison result according to res. If $res = 0$, $gain_i \geq gain_j$; Otherwise, $gain_i < gain_j$.

Correctness. The final comparison result res of two values of $[\![gain_i]\!]$ and $[\![gain_j]\!]$ is correct if:

- The comparison result satisfies $res = 0$, the original value of val is $val \geq 0$. We get

$$\begin{aligned} &[\![val]\!] = [\![\alpha_i^{\uparrow} \times \alpha_j^{\downarrow}]\!] \cdot [\![\alpha_i^{\downarrow} \times \alpha_j^{\uparrow}]\!]^{\eta-1} \\ &= [\![\alpha_i^{\uparrow} \times \alpha_j^{\downarrow} - \alpha_i^{\downarrow} \times \alpha_j^{\uparrow}]\!] \\ &\Rightarrow \alpha_i^{\uparrow} \times \alpha_j^{\downarrow} - \alpha_i^{\downarrow} \times \alpha_j^{\uparrow} \geq 0. \end{aligned}$$

Due to $\alpha_i^{\downarrow} \times \alpha_j^{\downarrow} > 0$, we get the derivation with the following formulas:

$$\begin{aligned} &gain_i - gain_j = \frac{\alpha_i^{\uparrow}}{\alpha_i^{\downarrow}} - \frac{\alpha_j^{\uparrow}}{\alpha_j^{\downarrow}} = \frac{\alpha_i^{\uparrow} \times \alpha_j^{\downarrow} - \alpha_i^{\downarrow} \times \alpha_j^{\uparrow}}{\alpha_i^{\downarrow} \times \alpha_j^{\downarrow}}, \\ &\Rightarrow gain_i - gain_j \geq 0, \\ &\Rightarrow gain_i \geq gain_j. \end{aligned}$$

- The comparison result satisfies $res = 1$, we get $val < 0$. It is derived from the following formulas:

$$\begin{aligned} &val = \alpha_i^{\uparrow} \times \alpha_j^{\downarrow} - \alpha_i^{\downarrow} \times \alpha_j^{\uparrow} < 0 \\ &\Rightarrow \alpha_i^{\uparrow} \times \alpha_j^{\downarrow} - \alpha_i^{\downarrow} \times \alpha_j^{\uparrow} < 0 \\ &\Rightarrow gain_i - gain_j < 0 \\ &\Rightarrow gain_i < gain_j. \end{aligned}$$

Based on above proof, we prove that if $val \geq 0$, then $\alpha_i^{\uparrow} \times \alpha_j^{\downarrow} - \alpha_i^{\downarrow} \times \alpha_j^{\uparrow} \geq 0$, *i.e.*, $gain_i - gain_j \geq 0$, $gain_i > gain_j$; Otherwise, $gain_i < gain_j$.

BuildGlobalNode: After receiving the locally optimal split $\{(\llbracket s_n^* \rrbracket, \llbracket f_n^* \rrbracket)\}_{n=1}^N$ and the evaluation indexes $\{\llbracket gain_n \rrbracket\}_{n=1}^N$ from N ENs, the CP will find the split $(\llbracket s^* \rrbracket, \llbracket f^* \rrbracket)$ with the maximal $gain$ to implement the global optimum.

Algorithm 2: Globally Optimal Split

Input: Encrypted gain parameters $\{\llbracket \alpha_n^{\uparrow} \rrbracket, \llbracket \alpha_n^{\downarrow} \rrbracket\}_{n=1}^N$, encrypted locally optimal split $\{\llbracket s_n^* \rrbracket, \llbracket f_n^* \rrbracket\}_{n=1}^N$.
Output: Globally optimal split f^* and s^*.

$\llbracket score^{\uparrow} \rrbracket \leftarrow \llbracket 0 \rrbracket$, $\llbracket score^{\downarrow} \rrbracket \leftarrow \llbracket 1 \rrbracket$;
$f^* \leftarrow \llbracket 0 \rrbracket$, $\llbracket s^* \rrbracket \leftarrow \llbracket 0 \rrbracket$;
for $0 < n \leq N$ **do**
  /* **CompareEncIndex** */
  $\llbracket score^{\uparrow} \times \alpha_n^{\downarrow} \rrbracket \leftarrow \mathtt{SMUL}(\llbracket score^{\uparrow} \rrbracket, \llbracket \alpha_n^{\downarrow} \rrbracket)$;
  $\llbracket score^{\downarrow} \times \alpha_n^{\uparrow} \rrbracket \leftarrow \mathtt{SMUL}(\llbracket score^{\downarrow} \rrbracket, \llbracket \alpha_n^{\uparrow} \rrbracket)$;
  $\mathtt{SCOM}(\llbracket score^{\uparrow} \times \alpha_n^{\downarrow} \rrbracket \cdot \llbracket score^{\downarrow} \times \alpha_n^{\uparrow} \rrbracket^{\eta-1}, \llbracket 0 \rrbracket)$;
  if $A - B < 0$ **then**
    $score^{\uparrow} \leftarrow \alpha_n^{\uparrow}$, $score^{\downarrow} \leftarrow \alpha_n^{\downarrow}$;
    $\llbracket f^* \rrbracket \leftarrow \llbracket f_n^* \rrbracket$, $\llbracket s^* \rrbracket \leftarrow \llbracket s_n^* \rrbracket$;
return $\llbracket f^* \rrbracket$, $\llbracket s^* \rrbracket$.

Computing with **CompareEncIndex**, CP can choose the globally optimal split $(\llbracket s^* \rrbracket, \llbracket f^* \rrbracket)$ for a decision node with the maximal $gain$ among submitted split $\{\llbracket s_n^* \rrbracket, \llbracket f_n^* \rrbracket\}_{n=1}^N$. Then, the globally optimal split $(\llbracket s^* \rrbracket, \llbracket f^* \rrbracket)$ will be sent to each EN as the split threshold of i-th decision node. The specific process of globally optimal split is shown in **Algorithm** 2. Once receiving the $(\llbracket s^* \rrbracket, \llbracket f^* \rrbracket)$, each EN builds the i-th decision node with the globally optimal split after decryption to obtain $s^* = \lfloor s^*/cons \rceil$. Then, the current sample space X is divided into leaf sub-space X_L and right sub-space X_R. After that, each EN will build the $(i+1)$-th tree node for locally optimal split. The process is iterated until reaching the tree height $\hbar$.

5.2.2 BuildLeaf

Upon reaching the tree height $\hbar$, the structure of k-th tree is fixed. To construct leaf nodes, the leaf weight of a tree is denoted as $w \in \mathbb{Z}^T$. The structure of a tree is represented as $q \in \mathbb{Z}^d \rightarrow \{1, 2, ..., T\}$, where T denotes the leaf number. The optimal weight in the j-th leaf is computed as

$$w_j^* = -\frac{G_j}{H_j + \psi}, \quad G_j = \sum_{i \in X_j} g_i, \; H_j = \sum_{i \in X_j} h_i,$$

where X_j is the sample space of leaf$_j$.

Therefore, the k-th tree is built in each EN, the tree structure is the same with other trees from ENs, but leaf nodes over local data are different from others. After K iterations, the XGBoost model XGB is constructed over each EN, which not only leverages shared knowledge over all ENs, but also remains local differences.

5.3 Secure Diagnosis on Edge

Considering the constraint of the limited computing capacity of MUs and the privacy of submitted-symptoms, we design a secure diagnosis strategy between the EN and MU.

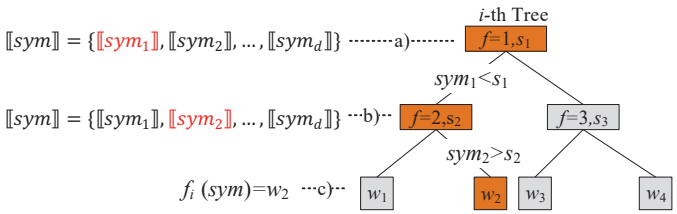


Fig. 5: The prediction on the k-th tree.

The whole process involves lightweight secure computation. Note that an EN owned by a medical institution stores the encrypted local diagnosis model for secure diagnosis over encrypted requests, and a MU owns mobile terminals and gives assistance for secure computation during the process of diagnosis. As the parameters of EN's trained model and information of MU's requests contain sensitive information, it is significant to provide strong privacy preservations without leaking any privacy during the diagnosis process.

Step 1: As the MU's characteristics contain sensitive information, the submitted-symptoms are computed as $\llbracket sym \rrbracket = \{\llbracket sym_1 \rrbracket, \llbracket sym_2 \rrbracket, ..., \llbracket sym_d \rrbracket\}$ with $\mathtt{Enc}$ algorithm. Besides, to prevent diagnosis result from being leaked to other MUs, the MU chooses a random number $r \leftarrow \mathbb{Z}_\eta$, and then submits $\llbracket r \rrbracket \leftarrow \mathtt{Enc}_{pk}(r)$ with $\llbracket sym \rrbracket$ and $\llbracket sym \rrbracket^{(2)} = \{\llbracket sym_1 \rrbracket^{(2)}, \llbracket sym_2 \rrbracket^{(2)}, ..., \llbracket sym_d \rrbracket^{(2)}\}$.

Step 2: Once receiving $\llbracket sym \rrbracket$, EN will implement secure diagnosis on an encrypted XGBoost as demonstrated in **Algorithm** 3, where $\mathtt{SCOM}$ is involved with the MU and EN to compare the submitted $\llbracket sym \rrbracket$ with the encrypted split threshold on each tree node from the root node until encountering a leaf node, where the split threshold of the current node is represented as $\llbracket s \rrbracket$ of f-th feature.

Taking the prediction on a tree as a toy example, Fig. 5 demonstrates the specific process: a) The threshold split of the root node is $\llbracket s_1 \rrbracket$ on the corresponding feature index $f = 1$, and $\llbracket sym_1 \rrbracket$ is the first feature of the submitted symptom $\llbracket sym \rrbracket$. Then, $\mathtt{SCOM}(\llbracket s_1 \rrbracket, \llbracket sym_1 \rrbracket)$ is executed to compare $\llbracket sym_1 \rrbracket$ with $\llbracket s_1 \rrbracket$. b) If the comparison result is $sym_1 < s_1$, then $\llbracket sym \rrbracket$ will be passed to the left child-node of the current node, whose threshold split is $\llbracket s_2 \rrbracket$ on the corresponding feature index $f = 2$. And also, $\llbracket sym_2 \rrbracket$ is the second feature value of $\llbracket sym \rrbracket$, and then $\llbracket sym_2 \rrbracket$ is compared with the split threshold $\llbracket s_2 \rrbracket$ by $\mathtt{SCOM}(\llbracket s_2 \rrbracket, \llbracket sym_2 \rrbracket)$. c) If the comparison result is $sym_2 > s_2$, then the right child-node of the current node is obtained, and its leaf weight is yielded as the prediction value $\mathcal{F}_k(sym) = w_2$.

In this way, after prediction on all K trees contained in the XGBoost, we obtain K leaf weights of each tree, then the final diagnosis results are $\llbracket \hat{y} \rrbracket = \llbracket \mathcal{F}_1(sym) \rrbracket \cdot \llbracket \mathcal{F}_2(sym) \rrbracket \cdot, ..., \cdot \llbracket \mathcal{F}_K(sym) \rrbracket$, *i.e.*, $\hat{y} = \sum_{k=1}^K \mathcal{F}_k(sym)$. Hence, the final diagnosis result $\llbracket \hat{y} \rrbracket$ is obtained. To protect the result from being leaked to other MUs, EN computes the returned diagnosis result $\llbracket \hat{y}' \rrbracket$ as $\llbracket \hat{y}' \rrbracket \leftarrow \llbracket \hat{y} \rrbracket \cdot \llbracket r \rrbracket$, $\llbracket \hat{y}' \rrbracket^{(1)} \leftarrow \mathtt{SDec}_{\lambda_1}(\llbracket \hat{y}' \rrbracket)$. Then, both $\llbracket \hat{y}' \rrbracket$ and $\llbracket \hat{y}' \rrbracket^{(1)}$ are sent to the MU.

Step 3: Once receiving the returned diagnosis result, the MU will decrypt the result with $\llbracket \hat{y}' \rrbracket^{(2)} \leftarrow \mathtt{SDec}_{\lambda_2}(\llbracket \hat{y}' \rrbracket)$, and $\hat{y}' \leftarrow \mathtt{WDec}(\llbracket \hat{y}' \rrbracket^{(1)}, \llbracket \hat{y}' \rrbracket^{(2)})$, and then remove the random number r to get the final result $\hat{y}$.

Correctness. The value of $\hat{y}$ is correct as: If $\hat{y}'$ satisfies

$$[\![\hat{y}']\!] = [\![\hat{y}+r]\!] = [\![\hat{y}]\!] \cdot [\![r]\!] = \varrho^{\hat{y}+r} \cdot r^{\eta} \mod \eta^2,$$
$$[\![\hat{y}']\!]^{(1)} = \mathtt{SDec}_{\lambda_1}([\![\hat{y}']\!]) = [\![\hat{y}']\!]^{\lambda_1} \mod \eta^2,$$
$$[\![\hat{y}']\!]^{(2)} = \mathtt{SDec}_{\lambda_2}([\![\hat{y}']\!]) = [\![\hat{y}']\!]^{\lambda_2} \mod \eta^2,$$
$$\hat{y}' = \mathtt{WDec}([\![\hat{y}']\!]^{(1)}, [\![\hat{y}']\!]^{(2)}) = \frac{\prod_{i=1}^{2} [\![\hat{y}']\!]^{(i)} \mod \eta^2 - 1}{\eta},$$

then the final prediction result $\hat{y}$ is yielded as $\hat{y} = \hat{y}' - r = (\hat{y}+r) - r$.

Algorithm 3: Secure Diagnosis on Encrypted XGBoost

Input: Encrypted instance $[\![sym]\!]$, an encrypted XGBoost $[\![\mathrm{XGB}]\!] = \{[\![\mathcal{F}_k]\!]\}_{k=1}^{K}$ comprises of K encrypted trees.
Output: Encrypted diagnosis result $[\![\hat{y}]\!]$.

```
[[ŷ]] ← [[0]];
for 1 ≤ k ≤ K do
    [[node]] ← the root node of [[F_k]];
    while node is a leaf node do
        if node is a leaf node then
            /* the k-th round prediction w */
            Obtain label weight [[w]] from the leaf node;
            /* ŷ = ŷ + w */
            [[ŷ]] ← [[ŷ]] · [[w]];
            break;
        else
            Obtain the split threshold [[s]] on f-th feature from [[node]];
            Obtain the f-th feature value [[sym_f]] from [[sym]];
            SCOM([[s]], [[sym_f]]); /* secure comparison SCOM */
            if s ≤ f then
                [[node]] ← [[node]].leftChild;
            else
                [[node]] ← [[node]].rightChild;
return [[ŷ]].
```

6 Security Analysis

In this section, we first give various attack types and then analysis the security of the proposed LPME system under these attacks.

6.1 Attaack Analysis

According to the descriptions in Section 4.2, we divide these attacks into the following types under the adversarial environment.

Type-I: Corruption : Assuming an adversary attempts to corrupt and collude CP, ENs and MUs to observe private information and tamper secret key stored in these entities. This type of attack consists of three attack models.

- *Ciphertext-only attack model*: The adversary can observe encrypted parameters and attempt to deduce secret keys.
- *Known-sample attack model*: The adversary can obtain some plaintext parameters with the corresponding ciphertexts and attempt to deduce secret keys.
- *Chosen-plaintext-attack model*: The adversary can encrypt certain plaintexts to obtain the corresponding ciphertexts for the deduction of secret keys.

Type-II: Eavesdropping: Assuming an adversary attempts to eavesdrop the transmitted information from the communication channel, he/she attempts to obtain privacy information and deduce these sensitive data.

6.2 Privacy Analysis

Based on the above-given attacks, we define the *real vs. ideal* model to formalize security analysis in LPME system. More specifically, assume that an adversary Adv interacts with a challenger in the real world to perform the predefined protocol $\prod$. Then, it interacts with a simulator Sim to complete the process in the ideal world. If the view of the adversary in the real-world is indistinguishable from the view in the ideal world, then we consider that LPME system is secure, which is represented as $\{\mathrm{IDEAL}_{\prod,\mathsf{Sim}}(m)\} \stackrel{c}{\equiv} \{\mathrm{REAL}_{\prod,\mathsf{Adv}}(m)\}$, where m is the input, $\prod$ is the corresponding protocol, and $\stackrel{c}{\equiv}$ denotes the computationally indistinguishable. We illustrate the security of LPME system as follows.

Theorem 1. Our proposed lightweight privacy-preserving machine learning is secure against semi-honest CP or ENs based on the semantic security of secure computation [31], which can resist the distinguishment of intermediate computational results with the non-collusion between CP and ENs.

Proof. Given the above *real vs. ideal* model, we separately analysis the security of each phase in our privacy-preserving XGBoost framework.

Assume that an adversary $\mathsf{Adv}_{\mathrm{CP}}$ corrupts CP. We construct a simulator $\mathsf{Sim}_{\mathrm{CP}}$ by executing in an ideal world, where all entities have trusted computation. The construction of $\mathsf{Sim}_{\mathrm{CP}}$ is executed as follows.

Phase 1: In each iteration of building a decision node over ENs, the i-th EN encrypts its local parameters $[\![gain_i]\!]$ and $([\![s_i^*]\!], [\![f_i^*]\!])$ to CP after running **BuildLocalNode** over individual training data. Then, $\mathsf{Sim}_{\mathrm{CP}}$ receives $\mathsf{Adv}_{\mathrm{CP}}$'s inputs $[\![gain_i]\!]$ and $([\![s_i^*]\!], [\![f_i^*]\!])$. Obviously, the semantic security of encrypted data has been proved to resist the *Type-II* attacks in the two-trapdoor cryptosystem. Since CP holds only one secret share $sk^{(2)}$, it cannot learn any content information from encrypted local parameters.

Phase 2: Then, $\mathsf{Sim}_{\mathrm{CP}}$ adopts **CompareEncIndex** to choose the globally optimal split over the encrypted parameters $\{([\![s_i^*]\!], [\![f_i^*]\!])\}_{i=1}^{N}$ from ENs. The secure computation (SMUL and SCOM) is involved in the process, and it has been proved be secure against the semi-honest adversary $\mathsf{Adv}_{\mathrm{CP}}$ [47], where $\mathsf{Adv}_{\mathrm{CP}}$ can corrupt CP in the real world. The view of $\mathsf{Adv}_{\mathrm{CP}}$ is defined as $\mathrm{REAL}_{\mathsf{Adv}_{\mathrm{CP}}} = \big(\{([\![s_i^*]\!], [\![f_i^*]\!])\}_{i=1}^{N}, \{[\![gain_i]\!]\}_{i=1}^{N}, sk^{(1)}, \{[\![gain_i^{(1)}]\!]\}_{i=1}^{N}\big)$, and $[\![gain_i]\!]^{(1)}$ is obtained with $\mathtt{SDec}_{sk^{(1)}}(\cdot)$ that still owns semantic security of the two-trapdoor cryptosystem. Given $\{[\![gain_i]\!]\}_{i=1}^{N}, sk^{(1)}$, $\mathsf{Sim}_{\mathrm{CP}}$ can simulate the view of $\mathsf{Adv}_{\mathrm{CP}}$ in the real world, the specific process is shown as follows:

- CP blinds random numbers to obtain $[\![gain_i]\!]'$, and uses $\mathtt{SDec}_{sk^{(1)}}(\cdot)$ to get decryption shares $[\![gain]\!]'^{(1)}$.

- The EN first implements $\mathtt{SDec}_{sk^{(2)}}(\cdot)$ and $\mathtt{WDec}(\cdot)$ algorithms to get $\overline{gain'}$ that contains blinded numbers, and then returns the corresponding computation result $[\![res]\!]$ to $\mathsf{Sim}_{\mathrm{CP}}$.

Informally speaking, if $\mathsf{Adv}_{\mathrm{CP}}$ distinguishes the view in the real world from the view in the ideal world with $sk^{(2)}$, then $\mathsf{Adv}_{\mathrm{CP}}$ is able to distinguish encrypted data or the decryption share, which violates semantic security of the two-trapdoor cryptosystem. Therefore, $\mathsf{Adv}_{\mathrm{CP}}$ is unfeasible to distinguish the ideal world from the real world. The distributions of $\mathrm{REAL}_{\mathsf{Adv}_{\mathrm{CP}}}$ and $\mathrm{IDEAL}_{\mathsf{Sim}_{\mathrm{CP}}} = \big(\{[\![gain_i]\!]\}_{i=1}^{N}, sk^{(1)}, \{\overline{[\![gain_i^{(1)}]\!]}\}_{i=1}^{N}\big)$ are indistinguishable with semantic security, as demonstrated in $\mathrm{IDEAL}_{\mathsf{Sim}_{\mathrm{CP}}} \overset{c}{\equiv} \mathrm{REAL}_{\mathsf{Adv}_{\mathrm{CP}}}$.

Besides, the adversary $\mathsf{Adv}_{\mathrm{EN}}$ can corrupt an EN, which includes the intermediate computational numbers $\{(gain_i^{\uparrow\prime}, gain_i^{\downarrow\prime})\}_{i=1}^{N}$. We construct a simulator $\mathsf{Sim}_{\mathrm{EN}}$ as follows:

- $\mathsf{Sim}_{\mathrm{EN}}$ simulates the process of **BuildLocalNode**. Given inputs $gain_i$ and (s_i^*, f_i^*), $\mathsf{Sim}_{\mathrm{EN}}$ receives $\mathsf{Adv}_{\mathrm{EN}}$'s inputs $gain_i'$ and returns encrypted data $[\![gain_i']\!]$ with random numbers r to $\mathsf{Adv}_{\mathrm{EN}}$.
- $\mathsf{Sim}_{\mathrm{EN}}$ sends $[\![gain_i']\!]$ to the trusted entity for honest **CompareEncIndex** computation. Then, $\mathsf{Sim}_{\mathrm{EN}}$ receives intermediate results $\{[\![gain_i^{(1)}]\!]\}$ with a secret share $sk^{(2)}$. At the same time, $\mathsf{Sim}_{\mathrm{EN}}$ returns intermediate results $\{\overline{[\![gain_i'^{(1)}]\!]}\}$ to $\mathsf{Adv}_{\mathrm{EN}}$.

Based on above analysis, the view of $\mathsf{Adv}_{\mathrm{EN}}$ is defined as $\mathrm{REAL}_{\mathsf{Adv}_{\mathrm{EN}}} = \big(sk^{(2)}, \{[\![gain_i']\!]\}_{i=1}^{N}, \{\overline{[\![gain_i'^{(1)}]\!]}\}_{i=1}^{N},\big)$. Given $\big(sk^{(2)}, \{[\![gain_i]\!]\}_{i=1}^{N}, \{[\![gain_i^{(1)}]\!]\}_{i=1}^{N}\big)$ to construct $\mathsf{Sim}_{\mathrm{EN}}$ with the above operations, the view of $\mathsf{Sim}_{\mathrm{EN}}$ is defined as $\mathrm{IDEAL}_{\mathsf{Sim}_{\mathrm{EN}}} = \big(\{[\![gain_i]\!]\}_{i=1}^{N}, \{[\![gain_i^{(1)}]\!]\}_{i=1}^{N}\big)$. Thus, we can conclude $\mathrm{IDEAL}_{\mathsf{Sim}_{\mathrm{EN}}} \overset{c}{\equiv} \mathrm{REAL}_{\mathsf{Adv}_{\mathrm{EN}}}$.

Therefore, *Type-I* attacks can be resisted. The proposed lightweight privacy-preserving XGBoost can resist *Type-I* and *Type-II* attacks which is unable to leak any privacy information. □

Theorem 2. Our secure diagnosis on edge is secure against the semi-honest EN or the MU as long as secure computation can resist the distinguishment of intermediate computational results and the collusion attack does not exist between ENs and MUs.

Proof. The specified analysis for secure diagnosis is shown as follows. The security requirement is resist an adversary $\mathsf{Adv}_{\mathrm{EN}}$ that corrupts the EN to obtain the privacy. Given the tuple of $(\lambda_1, [\![r]\!], [\![sym]\!])$, the simulator $\mathsf{Sim}_{\mathrm{EN}}$ is constructed as follows.

Phase 1: $\mathsf{Sim}_{\mathrm{EN}}$ runs **Algorithm** 3 on the encrypted submitted-symptoms $[\![sym]\!]$ from the MU to predict the diagnosis result $[\![\hat{y}]\!]$, the whole process involves secure comparison over two encrypted data (*i.e.,* SCOM), where the security of SCOM has been proved, more details are shown in [30]. During process of SCOM, there are no privacy leakage between the EN and the MU.

Phase 2: Besides, $[\![\hat{y}]\!]$ is obtained by combining with addition homomorphic. After that, the final diagnosis result is produced. To guarantee the confidentiality of diagnosis result, $\mathsf{Sim}_{\mathrm{EN}}$ first blinds the random number $[\![r]\!]$ with $[\![\hat{y}]\!]$ to output $[\![\hat{y}']\!]$ based on addition homomorphic without learning any information of encrypted data, then implements $[\![\hat{y}']\!]^{(1)} \leftarrow \mathtt{SDec}_{\lambda_1}(\overline{[\![\hat{y}']\!]})$ to get the decryption share, finally sends both $\overline{[\![\hat{y}']\!]}^{(1)}$ and $[\![\hat{y}']\!]$ to the MU. At the same time, the view of $\mathsf{Adv}_{\mathrm{EN}}$ is defined as $\mathrm{REAL}_{\mathsf{Adv}_{\mathrm{EN}}} = \big(\lambda_1, [\![\bar{r}]\!], [\![\overline{sym}]\!], [\![\bar{\hat{y}}]\!], [\![\overline{\hat{y}'}]\!], [\![\overline{\hat{y}'}]\!]^{(1)}\big)$.

Therefore, the distributions of $\mathsf{Adv}_{\mathrm{EN}}$ and $\mathsf{Sim}_{\mathrm{EN}}$ are unidentifiable with semantic security as shown in $\mathrm{IDEAL}_{\mathsf{Sim}_{\mathrm{EN}}} \overset{c}{\equiv} \mathrm{REAL}_{\mathsf{Adv}_{\mathrm{EN}}}$.

Assume that an adversary $\mathsf{Adv}_{\mathrm{MU}}$ corrupt a MU, the simulator $\mathsf{Sim}_{\mathrm{MU}}$ is constructed as follows.

- $\mathsf{Sim}_{\mathrm{MU}}$ simulates $[\![sym]\!] \leftarrow \mathtt{Enc}_{pk}(sym)$. And $\mathsf{Adv}_{\mathrm{MU}}$ runs on the same inputs to obtain $[\![\overline{sym}]\!]$ at the same time. Due to the semantic security of the two-trapdoor cryptosystem, the whole process can resist the *Type-II* attack.
- $\mathsf{Sim}_{\mathrm{MU}}$ simulates the prediction process with an EN, the details are similar to the execution process of $\mathsf{Sim}_{\mathrm{EN}}$.
- After obtaining the returned diagnosis result, $\mathsf{Sim}_{\mathrm{MU}}$ implements $[\![\hat{y}']\!]^{(2)} \leftarrow \mathtt{SDec}_{\lambda_2}([\![\hat{y}']\!])$ and $\hat{y}' \leftarrow \mathtt{WDec}([\![\hat{y}']\!]^{(1)}, [\![\hat{y}']\!]^{(2)})$ to get the blinded result $\hat{y}'$, and then removes the random number to obtain $\hat{y}$. At the same time, $\mathsf{Adv}_{\mathrm{MU}}$ outputs $\bar{\hat{y}}$.

Based on above analysis, itis inferred that the distributions of $\mathrm{REAL}_{\mathsf{Adv}_{\mathrm{MU}}}$ and $\mathrm{IDEAL}_{\mathsf{Sim}_{\mathrm{MU}}}$ are undistinguishable as shown in $\mathrm{IDEAL}_{\mathsf{Sim}_{\mathrm{MU}}} \overset{c}{\equiv} \mathrm{REAL}_{\mathsf{Adv}_{\mathrm{MU}}}$. Therefore, *Type-I* attacks can be resisted. Similar to the above analysis, *Type-I* and *Type-II* attacks can be resisted in the process of secure diagnosis. □

To guarantee the security of the LPME system, the malicious adversaries cannot distinguish encrypted data from the other encrypted data without the corresponding secret shares. Therefore, LPME system can resist *Type-I* and *Type-II* attacks with the above theorems.

7 Performance Analysis

In this section, we first detail the experiment setting, then analysis the theoretical performance compared with other privacy-preserving scheme [29]. Finally, we evaluate the performance of LPME system with two real-world datasets to illustrate the effectiveness, efficiency and feasibility.

7.1 Experimental Setting

We perform our evaluation on two public datasets.

- **Heart disease dataset**[2]: It consists of 303 instances, 14 features as well as two labels, where a patient without heart disease is labeled "0", a patient with heart disease is labeled "1".
- **Thyroid disease dataset**[3]: It contains 3,163 instances, 25 features and two labels, where a patient without thyroid disease is labeled "0", a patient with thyroid disease is labeled "1".

2. http://archive.ics.uci.edu/ml/datasets/Heart+Disease
3. http://www.kaggle.com/kumar012/hypothyroid

TABLE 3: Comparison between LPME and [29]

Overhead	Phase	LPME framework	Ma *et al.* [29]
Computational	Training	$\mathcal{O}(K2^{\hbar-1}N\mathtt{T_{SCOM}})$	$\mathcal{O}(t(2^{\hbar}-1)\mathtt{T_{total}})$
	Secure diagnosis	$\mathcal{O}(K2^{\hbar-1}\mathtt{T_{SCOM}})$	$\mathcal{O}(t(2^{\hbar}-1)\mathtt{T_{SCOM}})$
Communication	Training	$\mathcal{O}(NK2^{\hbar-1}\chi)$	$\mathcal{O}(\lvert X\rvert d\chi)$
	Secure diagnosis	$\mathcal{O}(K2^{\hbar-1}\chi_{\mathtt{SCOM}})$	$\mathcal{O}(t(2^{\hbar}-1)\chi_{\mathtt{total}})$

Notes. K denotes the number of trees contained in a XGBoost, t is the tree number of a random forest and χ denotes the bit length of an encrypted data. Besides, $\mathtt{T_{total}}$ are the total computation overhead of a decision node for secure computation in [29], and $\mathtt{T_{total}} = \epsilon_1\mathtt{T_{Enc}}+\epsilon_2\mathtt{T_{Add}}+\epsilon_3\mathtt{T_{SMUL}}+\epsilon_4\mathtt{T_{SCOM}}$, where $\mathtt{T_{Add}}$, $\mathtt{T_{SMUL}}$, $\mathtt{T_{SCOM}}$, $\mathtt{T_{Enc}}$, $\mathtt{T_{SDec}}$, $\mathtt{T_{WDec}}$ are the time for homomorphic addition, secure multiplication, secure comparison, Enc, SDec, and WDec, respectively. $|X|$ denotes the size of training data, $\chi_{\mathtt{total}}$ denotes the total communication overhead of a decision node for secure computation in [29], and $\chi_{\mathtt{total}} = \epsilon_3\chi_{\mathtt{SMUL}} + \epsilon_4\chi_{\mathtt{SCOM}}$, where ϵ_1, ϵ_2, ϵ_3 and ϵ_4 are operation times of Enc, homomorphic addition, secure multiplication and secure comparison, respectively.

Experiment setup: We use the cross-validation method to split 2/3 dataset as the training set and the remained as the validation set, our LPME system is implemented in Java, the experiments are evaluated on PC tester (3.30 GHz four-cores processors and 4 GB memory). To train a XGBoost model over multiple ENs, XGboost is adopted with the parameters $\gamma = 0$, $\psi = 20$ and sampling rate $rate_{sam} = 80\%$ over local training data of each EN.

For the construction of the k-th tree of a XGBoost, the whole process involves the construction of decision node with **BuildLocalNode** and **BuildGlobalNode** over N ENs. During the phase of **BuildLocalNode**, each EN builds local decision nodes described in Section 3. Then, multiple locally-built nodes are sent to CP for choosing the global optimal decision node with **BuildGlobalNode**. Finally, each EN builds individual leaf nodes over the local dataset with **BuildLeaf**.

7.2 Comparative Analysis

We implement detail comparison analysis of the computational overhead and communication overhead in Table 3 to demonstrate LPME system and [29]. Different from LPME system, [29] is a privacy-preserving distributed learning framework, which securely trains over encrypted training data.

Theoretical Analysis. For training the privacy-preserving machine learning, the computational complexity of [29] is $\mathcal{O}(t(2^{\hbar}-1)\mathtt{T_{total}})$, where all training processes of a decision node are implemented over encrypted data with secure computation, and $\mathtt{T_{total}} = \epsilon_1\mathtt{T_{Enc}}+\epsilon_2\mathtt{T_{Add}}+\epsilon_3\mathtt{T_{SMUL}}+\epsilon_4\mathtt{T_{SCOM}}$, ϵ_4 is a substantial number. This is because the large amount of SCOM are required to select the best split among numerous encrypted candidate values for training a decision node. On the other hand, the computational complexity of LPME framework is $\mathcal{O}(K2^{\hbar-1}N\mathtt{T_{SCOM}})$. Note that a large number of secure computation over encrypted data has been moved to plaintext computation, only several times SCOM are involved in the phase of **BuildGlobalNode**. Therefore, the computational overhead of training has been greatly reduced. Besides, the communication complexity of [29] is $\chi_{\mathtt{total}} = \epsilon_3\chi_{\mathtt{SMUL}} + \epsilon_4\chi_{\mathtt{SCOM}}$, only SCOM and SMUL are involved in the transformation of intermediate results, and the communication complexity of LPME framework is $\mathcal{O}(NK2^{\hbar-1}\chi)$. Since LPME framework involves the fewer number (*i.e.*, $NK2^{\hbar-1}$) of SCOM, there is less communication overhead in LPME framework compared with [29].

TABLE 4: Accuracy comparison

Tree number K		1	2	3	4	5
Heart dataset	LPME	80.4%	83.1%	87.0%	89.1%	90.6%
	XGBoost	78.9%	82.8%	86.8%	88.9%	90.3%
Thyroid dataset	LPME	89.3%	89.6%	94.8%	96.7%	97.1%
	XGBoost	89.5%	89.7%	95.0%	96.4%	97.0%

Notes. Tree height is $\hbar = 3$, $N = 3$, the size of samples is 100 items.

For the secure diagnosis, the computational complexity of LPME framework is $\mathcal{O}(K2^{\hbar-1}\mathtt{T_{SCOM}})$, and the communication complexity is $\mathcal{O}(K2^{\hbar-1}\chi_{\mathtt{SCOM}})$. In contrast, the computational complexity of [29] is $\mathcal{O}(t(2^{\hbar}-1)\mathtt{T_{SCOM}})$, and the communication complexity of [29] is $\mathcal{O}(t(2^{\hbar}-1)\chi_{\mathtt{total}})$.

Efficient Analysis. For training the privacy-preserving machine learning, LPME framework costs 5.567 s for the tree construction of each iteration, where the bit length is $|\eta| = 1024$ bits, tree height $\hbar = 3$, and the number of ENs is $N = 3$. In contrast, [29] costs 1.77×10^5 s for a single tree. Obviously, the efficiency has a significant enhancement in LPME framework.

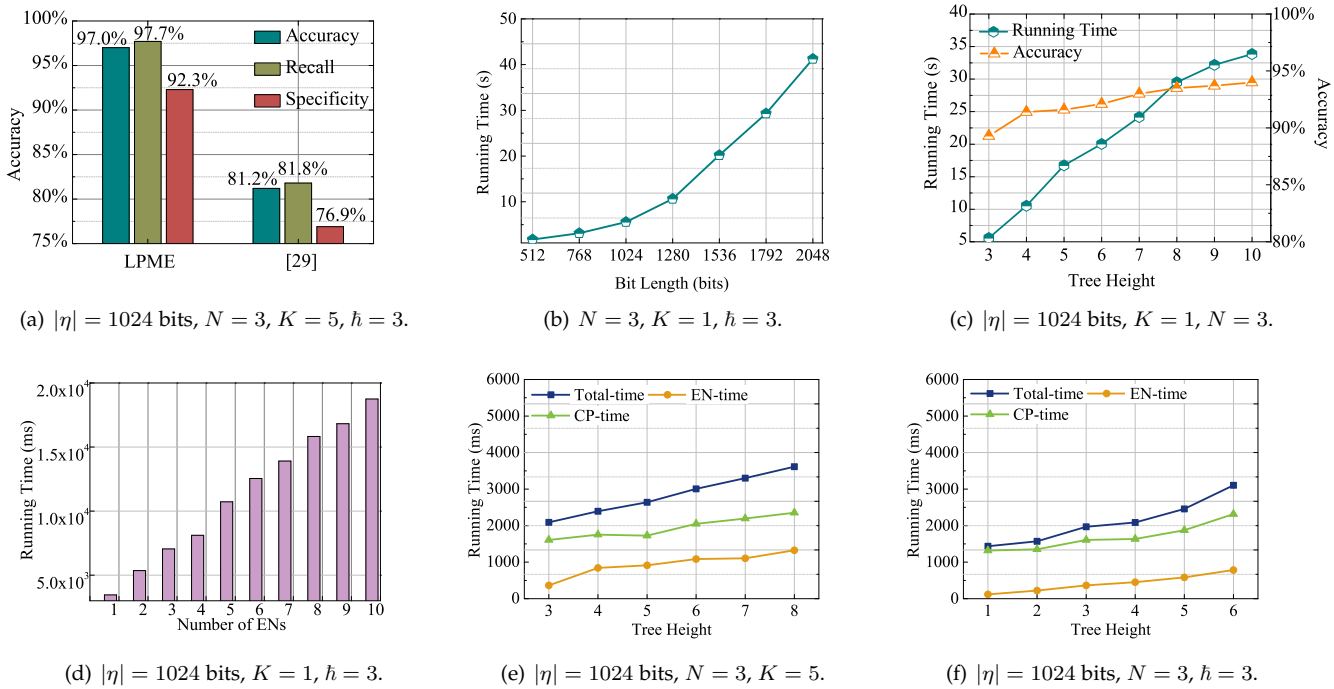
Effective Analysis. To measure the performance of machine learning, two indicators are introduced to measure the ability of predicting positive samples (*i.e.*, "recall") and predicting negative samples (*i.e.*, "specificity"). As represented in $recall = \frac{TP}{P_{total}}$ $specificity = \frac{FP}{F_{total}}$, TP denotes the positive samples whose prediction result is negative, FP denotes the negative samples whose prediction result is positive, P_{total} denotes the total number of positive samples, and F_{total} denotes the total number of negative samples.

As demonstrated in Fig. 6(a), the LPME system is compared with other privacy-preserving medical diagnosis schemes [29] over Thyroid disease dataset. When $|\eta| = 1024$ bits, $N = 3$, $K = 5$ and $\hbar = 3$, the accuracy is 97.0%, the recall is 97.7%, and the specificity is 92.3% in the LPME system, while the accuracy is 81.2%, the recall is 81.8% and 76.9% when $t = 5$ and $\hbar = 3$ in [29]. Therefore, the performance of LPME system has a significant improvement compared with [29].

7.3 Experimental Analysis

Effectiveness. For evaluating the accuracy of LPME system, we test the performance compared with the original XGBoost, as shown in Table 4. The observations are presented as follows:

- The accuracy of LPME system is improved with the increase of tree number K, the accuracy is 80.4% over Heart disease dataset and 89.3% over Thyroid disease dataset when $K = 1$, while the accuracy is 90.6% over Heart disease dataset and 97.1% over Thyroid disease dataset when $K = 5$.
- LPME system has a negligible difference of accuracy compared with the original XGBoost that implements over the global dataset, where the accuracy difference of two datasets is less than 1%. When $K = 5$, the accuracy is 90.6% (0.3% improvement) over Heart disease dataset and 97.1% (0.1% improvement) over Thyroid disease dataset compared with the original XGBoost.



(a) $|\eta| = 1024$ bits, $N = 3$, $K = 5$, $\hbar = 3$.

(b) $N = 3$, $K = 1$, $\hbar = 3$.

(c) $|\eta| = 1024$ bits, $K = 1$, $N = 3$.

(d) $|\eta| = 1024$ bits, $K = 1$, $\hbar = 3$.

(e) $|\eta| = 1024$ bits, $N = 3$, $K = 5$.

(f) $|\eta| = 1024$ bits, $N = 3$, $\hbar = 3$.

Fig. 6: Performance of LPME. (a) is the accuracy comparison of LPME and [29], where $|\eta| = 1024$ bits, 3 ENs, $K = 5$ and $\hbar = 3$. (b) is the running time of LPME system varying with bit length $|\eta|$, where 3 ENs, tree number $K = 1$ and tree height $\hbar = 3$. (c) is the running time and accuracy of LPME system varying with tree height $\hbar$, where $|\eta| = 1024$ bits, tree number $K = 1$ and 3 ENs. (d) is the running time of LPME system, where $|\eta| = 1024$ bits, tree number $K = 1$ and tree height $\hbar = 3$. (e) is the running time of private diagnosis strategy varying with tree height $\hbar \in [3, 8]$, where $|\eta| = 1024$ bits, 3 ENs and tree number $K = 5$. (f) is the running time of private diagnosis strategy varying with tree number $K \in [1, 6]$, where $|\eta| = 1024$ bits, 3 ENs and tree height $\hbar = 3$.

TABLE 5: Accuracy with the variation of tree height

Tree height $\hbar$	3	4	5	6	7	8
Heart dataset	80.4%	83.3%	87.6%	88.9%	92.1%	92.4%
Thyroid dataset	89.3%	89.5%	89.6%	91.5%	96.1%	97.1%

Notes. $K = 1$, the number of ENs is $N = 3$, the size of samples is 100 items.

Efficiency. From plotted in Figs. 6, we notice that the performance of LPME is influenced by the bit length of η, maximum tree height $\hbar$, and the size of ENs N, we evaluate the impact over the Thyroid disease dataset. As shown in Fig. 6(b), with the vary of bit length $|\eta|$, the running time of LPME system increases with the growth of bit length $|\eta|$. Since more ciphertexts are required to be processed, the running time grows when $|\eta|$ increases. To realize 80-bit security level[4], we denote $|\eta| = 1024$ bits in LPME system, where the running time of each iteration for training privacy-preserving machine learning costs 5.567 s, where EN number $N = 3$ and tree height $\hbar = 3$.

As demonstrated in Fig. 6(c), we observe that the accuracy and the running time of LPME system has an important influence with the vary of tree height $\hbar \in [3, 10]$. The reason is that with the tree grows, the higher the degree of fitting is, the smaller the training deviation (*i.e.*, higher accuracy) of the model is. Besides, as described in Table 5, we observe that the accuracy over two public datasets has the variation with varying tree height $\hbar \in [3, 8]$. However, it will lead to the overfitting problem when the tree height $\hbar$ grows too much. Moreover, with tree height $\hbar$ growing, there are involved more decision nodes to compute. Thus, the running time increases with the growth of $\hbar$. Considering both accuracy and running time of LPME system, we choose tree height $\hbar = 3$ to avoid the overfitting problem for training the privacy-preserving machine learning.

As represented in Fig. 6(d), we discover that with the growth of the number of ENs, the running time increases with the range of $N \in [1, 10]$, where each EN hosts 100 samples for training. The reason is that the more decision nodes are outsourced to construct the global model, it involves more ciphertexts in the **CompareEncIndex** algorithm for the construction of a global node. Thus, the running time increases with more ENs are joined.

For the secure diagnosis, the diagnosis time is shown in following: For an encrypted symptoms $[\![sym]\!]$, CP costs 2.03 s for diagnosis over encrypted data, and then MU costs 0.71 ms to decrypt the final result, where $K = 5$, $\hbar = 3$, $N = 3$, $|\eta| = 1024$ bits and $d = 25$. Since the diagnosis process involves ciphertext computation, the diagnosis time depends on the tree height $\hbar$. As plotted in Figs. 6(e)(f), we discover that the diagnosis time increases with the tree height $\hbar$ and tree number K. As the tree height $\hbar$ increases, the encrypted symptoms $[\![sym]\!]$ are required to compare with more decision nodes contained in a tree, the

4. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r4.pdf

corresponding feature value of $[\![sym]\!]$ is required to compare with encrypted split value of decision node. Thus, secure comparison SCOM on two encrypted data costs more time. Besides, the number of trees grows with each iteration. With more and more trees, the diagnosis result of the encrypted symptoms $[\![sym]\!]$ needs to implement over more trees. Therefore, the diagnosis time increases with tree number K.

8 Conclusion

This paper has proposed a lightweight privacy-preserving XGBoost framework on edge, which could not only provide lightweight XGBoost over edge nodes with strong privacy preservations, but also achieve privacy-preserving and real-time medical diagnosis on edge. The proposed LPME system with secure computation could securely construct XGBoost model with lightweight overhead, and efficiently provide medical diagnosis without privacy leakage. Experimental results over real-world datasets verified the efficiency and security of the LPME system on edge computing.

Acknowledgments

This work was supported by the Key Program of NSFC (No. U1405255), the Shaanxi Science & Technology Coordination & Innovation Project (No. 2016TZC-G-6-3), the National Natural Science Foundation of China (No. 61702404, No. 61702105, No. U1804263), the China Postdoctoral Science Foundation Funded Project (No. 2017M613080), the Fundamental Research Funds for the Central Universities (No. JB171504, No. JB191506), the National Natural Science Foundation of Shaanxi Province (No. 2019JQ-005).

References

[1] X. Wang, J. Ma, Y. Miao, X. Liu, and R. Yang, "Privacy-preserving diverse keyword search and online pre-diagnosis in cloud computing," *IEEE Transactions on Services Computing*, 2019.

[2] Y. Zhang, C. Xu, H. Li, K. Yang, J. Zhou, and X. Lin, "HealthDep: An efficient and secure deduplication scheme for cloud-assisted ehealth systems," *IEEE Trans. Industrial Informatics*, vol. 14, no. 9, pp. 4101–4112, 2018.

[3] B. Fu, P. Liu, J. Lin, L. Deng, K. Hu, and H. Zheng, "Predicting invasive disease-free survival for early-stage breast cancer patients using follow-up clinical data," *IEEE Transactions on Biomedical Engineering*, 2018.

[4] A. Galletta, L. Carnevale, A. Bramanti, and M. Fazio, "An innovative methodology for big data visualization for telemedicine," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 1, pp. 490–497, 2018.

[5] I. Kononenko, "Machine learning for medical diagnosis: history, state of the art and perspective," *Artificial Intelligence in medicine*, vol. 23, no. 1, pp. 89–109, 2001.

[6] C. P. Friedman, A. K. Wong, and D. Blumenthal, "Achieving a nationwide learning health system," *Science Translational Medicine*, vol. 2, no. 57, pp. 57cm29–57cm29, 2010.

[7] Y. Miao, X. Liu, K.-K. R. Choo, R. H. Deng, H. Wu, and H. Li, "Fair and dynamic data sharing framework in cloud-assisted internet of everything," *IEEE Internet of Things Journal*, 2019.

[8] Y. Miao, Q. Tong, K.-K. R. Choo, X. Liu, R. H. Deng, and H. Li, "Secure online/offline data sharing framework for cloud-assisted industrial internet of things," *IEEE Internet of Things Journal*, 2019.

[9] Y. Miao, J. Weng, X. Liu, K.-K. R. Choo, Z. Liu, and H. Li, "Enabling verifiable multiple keywords search over encrypted cloud data," *Information Sciences*, vol. 465, pp. 21–37, 2018.

[10] T. Ouyang, R. Li, X. Chen, Z. Zhou, and X. Tang, "Adaptive user-managed service placement for mobile edge computing: An online learning approach," in *Proc. IEEE Conference on Computer Communications (INFOCOM'19)*. IEEE, 2019, pp. 1468–1476.

[11] P. Dai, K. Liu, X. Wu, H. Xing, Z. Yu, and V. C. Lee, "A learning algorithm for real-time service in vehicular networks with mobile-edge computing," in *Proc. IEEE International Conference on Communications (ICC'19)*. IEEE, 2019, pp. 1–6.

[12] F. Wang, C. Zhang, J. Liu, Y. Zhu, H. Pang, L. Sun *et al.*, "Intelligent edge-assisted crowdcast with deep reinforcement learning for personalized qoe," in *Proc. IEEE Conference on Computer Communications (INFOCOM'19)*. IEEE, 2019, pp. 910–918.

[13] C.-C. Lin, D.-J. Deng, Y.-L. Chih, and H.-T. Chiu, "Smart manufacturing scheduling with edge computing using multi-class deep q network," *IEEE Transactions on Industrial Informatics*, 2019.

[14] G. S. Aujla, R. Chaudhary, K. Kaur, S. Garg, N. Kumar, and R. Ranjan, "Safe: Sdn-assisted framework for edge–cloud interplay in secure healthcare ecosystem," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 1, pp. 469–480, 2018.

[15] M. A. Sayeed, S. P. Mohanty, E. Kougianos, and H. P. Zaveri, "eseiz: An edge-device for accurate seizure detection for smart healthcare," *IEEE Transactions on Consumer Electronics*, 2019.

[16] A. M. Rahmani, T. N. Gia, B. Negash, A. Anzanpour, I. Azimi, M. Jiang, and P. Liljeberg, "Exploiting smart e-health gateways at the edge of healthcare internet-of-things: A fog computing approach," *Future Generation Computer Systems*, vol. 78, pp. 641–658, 2018.

[17] A. A. Ogunleye and W. Qing-Guo, "Xgboost model for chronic kidney disease diagnosis," *IEEE Transactions on Computational Biology and Bioinformatics*, 2019.

[18] M. Nishio, M. Nishizawa, O. Sugiyama, R. Kojima, M. Yakami, T. Kuroda, and K. Togashi, "Computer-aided diagnosis of lung nodule using gradient tree boosting and bayesian optimization," *PloS one*, vol. 13, no. 4, p. e0195875, 2018.

[19] A. R. Rao and D. Clarke, "A fully integrated open-source toolkit for mining healthcare big-data: architecture and applications," in *Proc. IEEE International Conference on Healthcare Informatics (ICHI'16)*. IEEE, 2016, pp. 255–261.

[20] E. Union, "General data protection regulation," 2018, https://gdpr-info.eu/.

[21] U. S. D. of Health and H. Services, "Health insurance portability and accountability act1996," 1996, https://www.hippa.com/.

[22] D. Liu, Z. Yan, W. Ding, and M. Atiquzzaman, "A survey on secure data analytics in edge computing," *IEEE Internet of Things Journal*, 2019.

[23] Y. Miao, X. Liu, R. H. Deng, H. Wu, H. Li, J. Li, and D. Wu, "Hybrid keyword-field search with efficient key management for industrial internet of things," *IEEE Transactions on Industrial Informatics*, 2018.

[24] Y. Miao, J. Ma, X. Liu, X. Li, Z. Liu, and H. Li, "Practical attribute-based multi-keyword search scheme in mobile crowdsourcing," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3008–3018, 2017.

[25] Y. Miao, X. Liu, K.-K. R. Choo, R. H. Deng, J. Li, H. Li, and J. Ma, "Privacy-preserving attribute-based keyword search in shared multi-owner setting," *IEEE Transactions on Dependable and Secure Computing*, 2019.

[26] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT '99)*. Springer, 1999, pp. 223–238.

[27] X. Liu, R. H. Deng, Y. Yang, H. N. Tran, and S. Zhong, "Hybrid privacy-preserving clinical decision support system in fog–cloud computing," *Future Generation Computer Systems*, vol. 78, pp. 825–837, 2018.

[28] S. Hu, M. Li, Q. Wang, S. S. Chow, and M. Du, "Outsourced biometric identification with privacy," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 10, pp. 2448–2463, 2018.

[29] Z. Ma, J. Ma, Y. Miao, and X. Liu, "Privacy-preserving and high-accurate outsourced disease predictor on random forest," *Information Sciences*, vol. 496, pp. 225–241, 2019.

[30] X. Liu, K.-K. R. Choo, R. H. Deng, R. Lu, and J. Weng, "Efficient and privacy-preserving outsourced calculation of rational numbers," *IEEE Transactions on Dependable & Secure Computing*, vol. 15, no. 1, pp. 27–39, 2018.

[31] X. Liu, R. H. Deng, K.-K. R. Choo, and J. Weng, "An efficient privacy-preserving outsourced calculation toolkit with multiple keys," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 11, pp. 2401–2414, 2016.

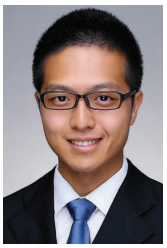
[32] Q. Wang, M. Du, X. Chen, Y. Chen, P. Zhou, X. Chen, and X. Huang, "Privacy-preserving collaborative model learning: The case of word vector training," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 12, pp. 2381–2393, 2018.

[33] P. Mohassel and Y. Zhang, "Secureml: A system for scalable privacy-preserving machine learning," in *Proc. IEEE Symposium on Security and Privacy (SP'17)*. IEEE, 2017, pp. 19–38.
[34] Y. Lindell and B. Pinkas, "Privacy preserving data mining," in *Proc. Annual International Cryptology Conference (CRYPTO'00)*. Springer, 2000, pp. 36–54.
[35] A. P. Sanil, A. F. Karr, X. Lin, and J. P. Reiter, "Privacy preserving regression modelling via distributed computation," in *Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'04)*. ACM, 2004, pp. 677–682.
[36] A. Fu, Z. Chen, Y. Mu, W. Susilo, Y. Sun, and J. Wu, "Cloud-based outsourcing for enabling privacy-preserving large-scale non-negative matrix factorization," *IEEE Transactions on Services Computing*, 2019.
[37] H. Yu, J. Vaidya, and X. Jiang, "Privacy-preserving svm classification on vertically partitioned data," in *Proc. Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'06)*. Springer, 2006, pp. 647–656.
[38] M. Al-Rubaie and J. M. Chang, "Privacy-preserving machine learning: Threats and solutions," *IEEE Security & Privacy*, vol. 17, no. 2, pp. 49–58, 2019.
[39] K. Cheng, T. Fan, Y. Jin, Y. Liu, T. Chen, and Q. Yang, "Secureboost: A lossless federated learning framework," *arXiv preprint arXiv:1901.08755*, 2019.
[40] T. Li, Z. Huang, P. Li, Z. Liu, and C. Jia, "Outsourced privacy-preserving classification service over encrypted data," *Journal of Network and Computer Applications*, vol. 106, pp. 100–110, 2018.
[41] Y. Aono, T. Hayashi, L. Wang, S. Moriai *et al.*, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1333–1345, 2017.
[42] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, "Beyond inferring class representatives: User-level privacy leakage from federated learning," in *Proc. IEEE Conference on Computer Communications (INFOCOM'19)*. IEEE, 2019, pp. 2512–2520.
[43] H. Zhang and K. Zeng, "Pairwise markov chain: A task scheduling strategy for privacy-preserving sift on edge," in *Proc. IEEE Conference on Computer Communications (INFOCOM'19)*. IEEE, 2019, pp. 1432–1440.
[44] J. Ni, K. Zhang, X. Lin, and X. S. Shen, "Securing fog computing for internet of things applications: Challenges and solutions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 601–628, 2017.
[45] Z. Ma, J. Ma, Y. Miao, K.-K. R. Choo, X. Liu, X. Wang, and T. Yang, "Pmkt: Privacy-preserving multi-party knowledge transfer for financial market forecasting," *Future Generation Computer Systems*, 2020.
[46] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'16)*. ACM, 2016, pp. 785–794.
[47] B. Pinkas, T. Schneider, N. P. Smart, and S. C. Williams, "Secure two-party computation is practical," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2009, pp. 250–267.

Zhuoran Ma received the B.E. degree from the School of Software Engineering, Xidian University, Xian, China, in 2017. She is currently a Ph.D candidate with the Department of Cyber Engineering, Xidian University. Her current research interests include data security and secure computation outsourcing.

Jianfeng Ma received the Ph.D. degree in computer software and telecommunication engineering from Xidian University, Xi'an, China, in 1988 and 1995, respectively. From 1999 to 2001, he was a Research Fellow with Nanyang Technological University of Singapore. He is currently a professor and a Ph.D. Supervisor with the Department of Computer Science and Technology, Xidian University, Xi'an, Chian. He is also the Director of the Shaanxi Key Laboratory of Network and System Security. His current research interests include information and network security, wireless and mobile computing systems, and computer networks.

Yinbin Miao received the B.E. degree with the Department of Telecommunication Engineering from Jilin University, Changchun, China, in 2011, and Ph.D. degree with the Department of Telecommunication Engineering from xidian university, Xi'an, China, in 2016. He is currently a Lecturer with the Department of Cyber Engineering in Xidian university, Xi'an, China. His research interests include information security and applied cryptography.



Ximeng Liu (S'13-M'16) received the B.Sc. degree in electronic engineering from Xidian University, Xian, China, in 2010 and the Ph.D. degree in Cryptography from Xidian University, China, in 2015. Now he is the full professor in the College of Mathematics and Computer Science, Fuzhou University. Also, he was a research fellow at the School of Information System, Singapore Management University, Singapore. He has published more than 200 papers on the topics of cloud security and big data security including papers in IEEE TOC, IEEE TII, IEEE TDSC, IEEE TSC, IEEE IoT Journal, and so on. He awards Minjiang Scholars Distinguished Professor, Qishan Scholars in Fuzhou University, and ACM SIGSAC China Rising Star Award (2018). His research interests include cloud security, applied cryptography and big data security. He is a member of the IEEE, ACM, CCF.

Kim-Kwang Raymond Choo (SM'15) received the Ph.D. in Information Security in 2006 from Queensland University of Technology, Australia. He currently holds the Cloud Technology Endowed Professorship at The University of Texas at San Antonio (UTSA). He is the recipient of various awards including the UTSA College of Business Col. Jean Piccione and Lt. Col. Philip Piccione Endowed Research Award for Tenured Faculty in 2018, ESORICS 2015 Best Paper Award. He is an Australian Computer Society Fellow, and an IEEE Senior Member.

Ruikang Yang received the BE degree from the School of Cyber Engineering, Xidian University, Shannxi, China, in 2017. He is currently working towards the PhD degree in the Xidian University. His research interest includes data security and cloud computing security.

Xiangyu Wang currently is a Ph.D candidate in Xidian University. He received the B.E. degree with the School of Cyber Engineering from Xidian University, Shannxi, China, in 2017. His research interests include data security and secure computation outsourcing.