Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 7746286, 13 pages
https://doi.org/10.1155/2021/7746286

Research Article

An Efficient Online Multiparty Interactive Medical Prediagnosis Scheme with Privacy Protection

Qiuyue Zhang,[1,2] **Xiao Zheng**,[1,2,3] **and Xiujun Wang**[1,2]

[1]*College of Computer Science and Technology, Anhui University of Technology, Maanshan, Anhui 243032, China*
[2]*Anhui Engineering Laboratory for Intelligent Applications and Security of Industrial Internet, Maanshan, Anhui 243032, China*
[3]*Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei, Anhui 230071, China*

Correspondence should be addressed to Xiao Zheng; xzheng@ahut.edu.cn

Received 1 September 2021; Accepted 2 November 2021; Published 15 December 2021

Academic Editor: Yushu Zhang

Medical prediagnosis systems are now available online to give users quick and preliminary diagnosis information. The need for such a system has become particularly evident in areas with insufficient health professionals. Due to the privacy of patient medical information and the sensitivity of cloud diagnosis models, it is necessary to protect the security of data, models, and communications. These existing diagnosis systems can hardly provide a satisfied diagnosis accuracy while ensuring comprehensive security and high efficiency. In order to solve these problems, we proposed Relief-*k* minimum Wasserstein distance (Relief-*k*MW) classification method, which combined data encryption and BLS signature to form a privacy-preserving efficient online multiparty interactive medical prediagnostic scheme (OMPD). Theoretical analysis shows our OMPD effectively provides high-precision prediagnosis services. Extensive experimental results demonstrate that OMPD not only greatly improves the diagnostic accuracy but also reduces the computational and communication overhead.

1. Introduction

With the rapid development of mobile Internet, wearable devices, and intelligent Internet of Things, online medical prediagnosis systems that can provide prediagnosis services and medical advice anytime and anywhere have received extensive research attention due to their importance. Typically, an online medical prediagnosis system needs to provide a high degree of diagnostic accuracy along with a strong level of privacy protection. In order to achieve these two goals at the same time or at least maintain an acceptable balance between the two goals, multiple factors need to be considered. Firstly, a good classifier must be carefully selected from the existing classification algorithm library to achieve high diagnostic accuracy. And the classifier must be refined to capture the peculiar nature of the online diagnostic problem. The existing literature already provides us some candidate algorithms such as random forest [1], neural network [2], and other methods [3–5], and they have been widely used in medical diagnosis. However, in general, these algorithms often yield either an interminable training and response time or an unbearably low diagnostic accuracy and privacy protection. Therefore, we need to reinvestigate the potential of existing classifiers and then select the one that can be modified to offer the experimentally best performances in terms of computational overhead and privacy protection.

Secondly, in order to keep the medical data used by an online medical prediagnosis system well protected from malicious users who may attach and profit from these data, we need to choose appropriate privacy protection schemes [6–15] for security needs. There are three kinds of privacy protection schemes: anonymity protection, differential privacy protection, and homomorphic encryption. The anonymity protection (such as *k*-anonymity [9] and *l*-diversity [10]) simply erases users' private information which may lead to a decreased diagnostic accuracy as the deleted data cannot be used. The differential privacy protection [11–13] protects users' privacy by adding noise to completely obfuscate the query response. However, these random noise can cover the critical information that are needed to boost the diagnostic accuracy. Different from the prior two schemes, homomorphic encryption [14, 15] can strictly protect privacy without destroying the original data. However, the encryption and decryption process of the homomorphic encrption usually requires huge computational overhead. Therefore, to have homomorphic encryption work for the scenarios of online medical prediagnosis, it still requires us considerable effort to bring the computational burden of the homomorphic encryption down considerably without sacrificing the protection strength very much.

In addition, a multiparty interactive system usually needs to be aware of the problem of communication security. This becomes particularly essential for online medical prediagnosis scenarios with vulnerable and unsecure communication channels. It is possible that the transmitted messages between the two parties can be eavesdropped malicious attackers. Hence, we need an efficient information security scheme for online medical prediagnosis system to protect the communication. For this target, there are some existing methods [16–19], such as secure multiparty protocol, digital signature, and some other data encryption methods. In this paper, we choose digital signature information authentication scheme. Because it can ensure that only the correct recipient can obtain the communication information through its private key, it can effectively ensure that the communication content is not maliciously stolen or tampered with. The BLS short signature [16] is considered to be one of the most effective methods. This technology authenticates and confirms users' identity information. It can prevent others from fraudulently using users' identity information.

Based on the above three points, it is difficult to combine the classification method with the privacy protection method, while saving computation and communication overhead to achieve high system efficiency. In this paper, we proposed an online multiparty interactive medical prediagnosis service scheme with high efficiency, high precision, and privacy protection, called OMPD. It can protect the private information of medical users, a large of medical instances of the hospital and the diagnosis model in the cloud. And the users can obtain online medical prediagnosis services when the original data is not available in the cloud. The main contributions of this paper are as follows:

(1) OMPD can provide high-precision prediagnosis services. We introduced a data preprocessing method (simple data encryption and some conversion) based on the newly proposed Relief-*k*MW classifier, and these data processing operations would not change the original classification accuracy of the classifier. In order to verify the classification accuracy of OMPD, we conducted accuracy analysis and experiments on two real data sets on the UCI machine learning library (http://archive.ics.uci.edu/ml). Experimental results showed that OMPD can provide high-precision services

(2) OMPD is a three-party interactive system that can provide medical prediagnosis services with full-process protection. By preprocessing medical data and applying BLS signatures to communication information, the security of private information of medical users, the instance of the hospital's database, the diagnostic model in the cloud, and the interaction can be guaranteed

The structure of our work is as follows: Section 2 introduces some related works, and Section 3 introduces the Relief-*k*MW classification method and BLS signatures. Section 4 introduces the entire detailed process of the OMPD. Section 5 carries out accuracy and safety analysis. The experimental evaluation is carried out in Section 6. Finally, we conclude in Section 7.

2. Related Works

In recent years, more and more people have paid attention to the efficiency and privacy safety of medical diagnosis services, and many solutions [20–25] have been proposed. In view of the privacy and security issues in online medical prediagnosis services, homomorphic encryption technology can well protect the private information in medical data and is widely used in various medical diagnosis schemes. For example, literature [22] constructed three classification protocols based on the Paillier cryptosystem to protect the security of data collected from medical users and service providers. Literature [23] developed an automatic diagnosis system for privacy protection. The remote server classified the biomedical signal provided by the client without obtaining any information about the signal itself and the final result of the classification. Liu et al. [24] proposed a privacy-preserving patient-centered clinical decision support system based on additive homomorphism to help clinicians assist in diagnosing patients' disease risks. Hua et al. [25] proposed an efficient and privacy-preserving medical diagnosis framework, which outsourced the accurate diagnosis model to a cloud server in an encrypted manner based on partial decryption and security comparison technology to achieve the advantages of two-way ciphertext quantification. Since all encryption operations are based on homomorphic encryption, the huge computational overhead makes the system extremely inefficient, which is not suitable for online medical service scenarios.

In addition, many online medical prediagnosis schemes based on various machine learning classification algorithms [26–33] adopt different privacy protection strategies. Wu et al. [27] designed a new efficient and privacy-preserving conditional unintentional transmission protocol. The literature [28] proposed a novel privacy protection biometric identification scheme, which improved efficiency by using the power of cloud computing. Zhang et al. [29] proposed a cloud-based privacy protection deep computing model to improve the efficiency of big data feature learning. Their schemes mainly protect the privacy information of users' query vector, but do not protect the security of communication information and diagnostic models.

Zhang et al. [32] proposed a disease prediction system (called PPDP) that used random matrices to construct new medical data encryption, disease learning, and disease prediction. Although the use of a single-layer perceptron makes the disease prediction stage simple and efficient, the accuracy of the prediagnosis is not high enough. And in the disease learning stage, constantly updating the weights until convergence would consume huge computation and communication overhead. At the same time, in this three-party interaction scenario, the users' private information is directly transmitted without communication security. Zhu et al. [33] proposed an efficient and privacy-preserving medical primary diagnosis based on *k*NN (called EPDK). With lightweight multiparty random shielding and polynomial aggregation technology, users can ensure the security of their sensitive information in the online medical diagnosis. This is an interactive service in which only the user and the server participate. The diagnostic model of the server is based on the original medical data set. Medical data that is not encrypted or processed is vulnerable to attack or theft. In addition, the classification accuracy of EPDK is still not high enough. There are few schemes that provide comprehensive privacy protection, high-precision, and high-efficiency prediagnosis services.

3. Preliminaries

This section introduces the Relief-*k*MW classifier used by OMPD and the BLS signature technology to protect communication security.

3.1. Relief-kMW Classifier. The Relief-*k*MW classifier first needs to use the Relief algorithm [34] to obtain $weight = (w_1, w_2, \cdots, w_n)$, and then calculates the Relief-Wasserstein distance between the query vector $X = (x_1, x_2, \cdots, x_n)$ and each medical instance $Y^{(j)} = (y_1^{(j)}, y_2^{(j)}, \cdots, y_n^{(j)})$, $j \in [1, N]$ in the database D, where n is the number of features in each vector, and N is the total number of data in the database. The specific steps are as follows:

3.1.1. Relief Feature Weight Distribution Algorithm. Before the Relief-*k*MW classifier works, the weight of the feature is calculated according to the Relief feature weight distribution algorithm (as shown in Algorithm 1).

The function $\mathrm{diff}(i, R, H)$ in Algorithm 1 represents the difference between the sample R and the sample H on the i -th feature, and its formula is as follows:

$$\mathrm{diff}(i, R, H) = \begin{cases} 0, & \text{if } f_i \text{ discrete and } R[i] = H[i] \\ \dfrac{R[i] - H[i]}{\max(i) - \min(i)}, & \text{if } f_i \text{ continuous} \\ 1, & \text{if } f_i \text{ discrete and } R[i] \neq H[i] \end{cases}. \quad (1)$$

3.1.2. Relief-kMW Classification Method. After getting the weight, we perform the Relief-*k*MW classification method, which is similar to *k*NN, except that *k*NN uses Euclidean distance, and our method uses Relief-Wasserstein distance. First, we calculate the Relief-Wasserstein distance between the query vector $X = (x_1, x_2, \cdots, x_n)$ and each instance $Y^{(j)} = (y_1^{(j)}, y_2^{(j)}, \cdots, y_n^{(j)})$, $j \in [1, N]$ in the database D, its definition as the following:

$$\begin{aligned} \theta_0 &= 0, \\ \theta_1 &= \theta_0 + w_1 \times (x_1 - y_1), \\ \theta_2 &= \theta_1 + w_2 \times (x_2 - y_2), \\ &\cdots \\ \theta_n &= \theta_{n-1} + w_n \times (x_n - y_n). \end{aligned} \quad (2)$$

Definition 1 (Relief-Wasserstein distance). Suppose $X = (x_1, x_2, \cdots, x_n)$ and $Y = (y_1, y_2, \cdots, y_n)$ are two input samples, where n is the total number of features, θ_i is the iterative value of the difference between X and Y on each component, and $weight = (w_1, w_2, \cdots, w_n)$ is the weight value of the corresponding features obtained by the Algorithm 1. Then, we calculate the followings:

The Relief-Wasserstein distance between X and Y is

$$RW = \sum_{i=1}^{n} |\theta_i|. \quad (3)$$

Then, we select the closest k instances. At last, we use most of the classification results of these k instances as the final classification result. In fact, the choice of k value is not optimal in theory. It depends on data characteristics and classification requirements. Some articles [35] on the optimal theoretical value of k pointed out that the best choice of k for a given data set may also depend on many attributes of the data. They have carried out the selection experiment of k value under different applications for different specific data sets and selected the best k value for the application as much as possible.

3.2. BLS Signature. Suppose there is a large prime number q and two cyclic groups G_1 and G_2, their orders are both q, and g is a generator of G_1. Then, there is a mapping $e : G_1 \times G_1 \longrightarrow G_2$, for $\forall a, b \in Z_q^\star$ and $\forall u, v \in G_1$, and it has the following properties:

$$e\left(g^a, g^b\right) = e\left(g^b, g^a\right) = e(g, g)^{ab}. \quad (4)$$

(1) Generate public key

Definition 2 (BLS signature). The bilinear parameter generator takes a safety parameter μ as input and outputs a 5-tuple (q, g, G_1, G_2, e), where q is the prime number of μ, G_1 and G_2 are two cyclic groups of order q, and g is the generator of G_1, $e : G_1 \times G_1 \longrightarrow G_2$ is a nondegenerate and effectively calculated bisexual mapping. The steps of BLS signature are as follows:

The message sender randomly selects an integer as the private key SK and $0 < \mathrm{SK} < q - 1$, and then calculates the

```
**Input:** training data set $D$, sample sampling times $m$, feature set $F = (f_1, f_2, \cdots, f_n)$, which has n features in total.
**Output:** feature weight weight $= (w_1, w_2, \cdots, w_n)$.
The feature rights are reset to 0;
for $r = 1$ to $m$ do
    Randomly select a sample $R$;
    Find the nearest neighbor sample $H$ in the same class of $R$, and the nearest neighbor sample $M$ in different classes of $R$;
    for $i = 1$ to $n$ do
        $w(i) = w(i) - (\text{diff}(i, R, H)/m) + (\text{diff}(i, R, M)/m)$;
    end for
end for
return weight.
```

Algorithm 1: Relief feature weight distribution algorithm.

public key:

$$\text{PK} = \text{g}^{\text{SK}}. \quad (5)$$

(2) Create a signature. The sender creates a signature by performing the following operations on the message $m \in G_1$:

$$\text{Sig} = m^{\text{SK}}. \quad (6)$$

(3) Verify the signature. After receiving the signature, the receiver performs the verification of the following formula, and the message content can be obtained after the verification is successful:

$$e(g, \text{Sig}) \overset{?}{=} e(\text{PK}, m). \quad (7)$$

4. OMPD Scheme

Our OMPD includes three entities: the hospital, the cloud, and the users. And it consists of six stages: initialization, query generation, query processing, prediagnosis service, query result analysis, and result acquisition. Figure 1 shows the flow of the OMPD. For ease of expression, Table 1 gives the description of the notations used in the following sections.

4.1. Initialization. The hospital first generates a bilinear parameter (q, g, G_1, G_2, e). Then, the hospital uses a random number as private key SK_H ($\text{SK}_H \in Z_q^\star$), calculates public key $\text{PK}_H = g^{\text{SH}_H}$, and sets the parameters t_1, t_2, and t_3, where $t_1 > t_2 > t_3$. It needs to choose a secure asymmetric encryption algorithm $E()$ and encrypted hash function Hash(), where Hash(): $\{0, 1\}^\star \longrightarrow G_1$. The hospital securely saves its private key SK_H as the master key and publishes system parameters $(q, g, G_1, G_2, e, t_1, t_2, t_3, E(), \text{Hash}())$.

Suppose there is a medical data set $\{D_l = \{Y_l^{(j)} = (y_{l1}^{(j)}, y_{l2}^{(j)}, \cdots, y_{ln_l}^{(j)}), j \in [1, N_l]\}, l \in [1, L]\}$ in the hospital database, these medical instances include L kinds of diseases, and each disease corresponds to a data subset D_l containing N_l data. For each D_l, the instance contains n_l features. The hospital uses the Algorithm 1 to obtain the feature weight $\text{weight}_\text{l} = (w_{l1}, w_{l2}, \cdots, w_{ln_l})$ in each data subset D_l. The preprocessing of medical data is shown in Algorithm 2. The hospital selects two large prime numbers p and a and sets $\text{len}(p) = \text{len}(t_1)$, $\text{len}(a) = \text{len}(t_2)$. Next, it chooses a large random number β and $\beta \in Z_p$. Each medical instance $Y_l^{(j)} = (y_{l1}^{(j)}, y_{l2}^{(j)}, \cdots, y_{ln_l}^{(j)}), j \in [1, N_l]$ should successively perform vector feature weight distribution (to get the vector $IY_l^{(j)} = (Iy_{l1}^{(j)}, Iy_{l2}^{(j)}, \cdots, Iy_{ln_l}^{(j)}), j \in [1, N_l]$), vector value iterative transformation (to get the vector $IIY_l^{(j)} = (IIy_{l1}^{(j)}, IIy_{l2}^{(j)}, \cdots, IIy_{ln_l}^{(j)}), j \in [1, N_l]$), forward expansion (to get the vector $IIIY_l^{(j)} = (IIIy_{l1}^{(j)}, IIIy_{l2}^{(j)}, \cdots, IIIy_{ln_l+2}^{(j)}), j \in [1, N_l]$), and finally to get the processed medical vector $EY_l^{(j)} = (ey_{l1}^{(j)}, ey_{l2}^{(j)}, \cdots, ey_{ln_l+2}^{(j)}), j \in [1, N_l]$. The time complexity of Algorithm 2 is $O(n_l)$.

The hospital keeps the parameters (p, a, b_i^j) secret and sends all preprocessed medical instances to the cloud. After obtaining the preprocessed medical instances, the cloud randomly selects part of the data as the test set to obtain the optimal k_l value of the classifier for the diseases l. Then, the cloud can have the Relief-kMW classifier with the best value k_l and many preprocessed medical instances received from the hospital.

4.2. Query Generation. The user generates query vector $X = (x_1, x_2, \cdots, x_{n_l})$. Then, the user uses the hospital's public key PK_H to encrypt the query vector to get $X_Q = E_{\text{PK}_H}(l\|x_1\|x_2\|\cdots\|x_{n_l})$, where l is the disease that the user wants to query. Then, he (she) uses private key SK_U to create a signature $\text{Sig}_U = \text{Hash}(X_Q\|TS_1)^{\text{SK}_U}$ and then sends $ID_U\|X_Q\|\text{Sig}_U\|TS_1$ to the hospital.

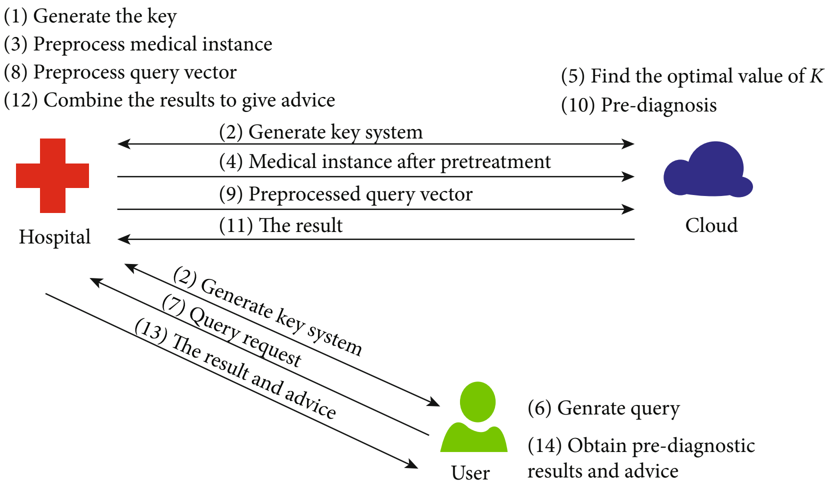
(1) Generate the key
(3) Preprocess medical instance
(8) Preprocess query vector
(12) Combine the results to give advice
(5) Find the optimal value of K
(10) Pre-diagnosis
(2) Generate key system
(4) Medical instance after pretreatment
(9) Preprocessed query vector
(11) The result
Hospital
Cloud
(2) Generate key system
(7) Query request
(13) The result and advice
(6) Genrate query
(14) Obtain pre-diagnostic results and advice
User

Figure 1: Architecture diagram of OMPD.

4.3. Query Processing. After receiving the $ID_U\|X_Q\|Sig_U\|TS_1$ from the user, the hospital first needs to confirm the ID_U, and then uses the following formula to verify the validity of the message:

$$e(g, Sig_U) \overset{?}{=} e\left(PK_U, Hash\left(X_Q\|TS_1\right)\right). \tag{8}$$

If the equation is true, then the hospital performs Algorithm 2. The processing method is the same as that of medical instances. The hospital selects a large random number γ, and $\gamma \in Z_p$ then performs vector features weight distribution (to get the vector $IX = (Ix_1, Ix_2, \cdots, Ix_{n_l})$), vector value iterative transformation (to get the vector $IIX = (IIx_1, IIx_2, \cdots, IIx_{n_l})$), and two-dimensional forward expansion (to get the vector $IIIX = (IIIx_1, IIIx_2, \cdots, IIIx_{n_l+2})$), and obtains the preprocessed query vector $EX = (ex_1, ex_2, \cdots, ex_{n_l+2})$ after encryption.

The hospital calculates $Q = E_{\mathrm{PK}_C}(l\|\beta\|\gamma\|ex_1\|ex_2\|\cdots\|ex_{n_l+2})$ and keeps the parameters (p, a, b_i) secret. Then, it uses private key SK_H to create a signature $\mathrm{Sig}_H = \mathrm{Hash}(Q\|TS_2)^{SK_\mathrm{H}}$ and sends $Q\|\mathrm{Sig}_H\|TS_2$ to the cloud.

4.4. Prediagnostic Service. After receiving $Q\|Sig_H\|TS_2$ from the hospital, the cloud verifies the validity of the message by using the following formula:

$$e(g, \mathrm{Sig}_H) \overset{?}{=} e(\mathrm{PK}_H, \mathrm{Hash}(Q\|TS_2)). \tag{9}$$

If the above equation is true, the cloud performs the prediagnosis service algorithm. As shown in Algorithm 3, the cloud finds the data subset D_l corresponding to the disease l and calculates the Relief-Wasserstein distance between EX and each preprocessed encrypted medical instance in D_l. Then, the cloud selects the k_l instances with the smallest distance and takes most of the categories of these k_l instances as the final prediagnosis result R. The time complexity of Algorithm 3 is $O(N_l n_l)$. The cloud keeps each RW, and computation rules secret. Then, it uses the hospital's public key to calculate $R_l = E_{PK_H}(R)$ and uses SK_C to create a signature $\mathrm{Sig}_C = \mathrm{Hash}(R_l\|TS_3)^{\mathrm{SK}_\mathrm{C}}$. Findly, the cloud sends $R_l\|\mathrm{Sig}_C\|TS_3$ to the hospital.

4.5. Query Result Analysis. After receiving $R_l\|Sig_C\|TS_3$ from the cloud, the hospital verifies the validity of the message by using the following formula:

$$e(g, Sig_C) \overset{?}{=} e(PK_C, Hash(R_l\|TS_3)) \tag{10}$$

The prediagnosis result R can be obtained if the equation holds. The hospital gives some advice A_l for this result and calculates $RA_l = E_{PK_U}(R\|A_l)$. Then, the hospital uses private key to create a signature $\mathrm{Sig}_H = \mathrm{Hash}(\mathrm{RA}_1\|\mathrm{TS}_4)^{SK_H}$ and sends $\mathrm{RA}_1\|\mathrm{Sig}_H\|\mathrm{TS}_4$ to the user.

4.6. Result Acquisition. After receiving $\mathrm{RA}_1\|\mathrm{Sig}_H\|\mathrm{TS}_4$ from the hospital, the user verifies the validity of the message by using the following formula:

$$e(g, \mathrm{Sig}_H) \overset{?}{=} e(\mathrm{PK}_H, \mathrm{Hash}(\mathrm{RA}_1\|\mathrm{TS}_4)) \tag{11}$$

If the equation holds, the user can obtain the prediagnosis result R and the advice A_l.

5. Accuracy and Security Analysis

This section analyzes the accuracy and security of OMPD. We verify that the data preprocessing of OMPD does not affect the original classification accuracy and can ensure privacy security.

5.1. Accuracy Analysis. If it can be verified that the Relief-Wasserstein distance RW_2 between the preprocessed data calculated by Algorithm 3 and the Relief-Wasserstein distance RW between the original data are approximately a fixed multiple, it can be proved that our scheme does not

Table 1: Descriptions of notations.

Notation	Description
t_1, t_2, t_3	Safety parameters selected by the hospital
(q, g, G_1, G_2, e)	Parameters of the bilinear group
$E(), \mathrm{Hash}()$	Secure asymmetric encryption algorithm and cryptographic hash function
$\mathrm{PK}_H, \mathrm{PK}_U, \mathrm{PK}_C$	Public keys for the hospital, the user, and the cloud
$\mathrm{SK}_H, \mathrm{SK}_U, \mathrm{SK}_C$	Secret keys for the hospital, the user, and the cloud
$Sig_H, \mathrm{Sig}_U, \mathrm{Sig}_C$	Signatures created by the hospital, the user, and the cloud
L, l	Total number of disease labels and the lth disease label
D_l, N_l	The data subset to disease l, the total number of data in D_l
n_l	The number of features contained in each data in D_l
$\mathrm{weight}_l, w_{li}$	The feature weight vector and the ith component of weight_l
$X, Y_l^{(j)}$	Query vector and the jth data in D_l
$\mathrm{I}X, \mathrm{I}Y_l^{(j)}$	The vector obtained by feature weight distribution of $X, Y_l^{(j)}$
$\mathrm{II}X, \mathrm{II}Y_l^{(j)}$	The vector obtained by iterative transformation of $IX, IY_l^{(j)}$
$\mathrm{III}X, \mathrm{III}Y_l^{(j)}$	The vector obtained by forward expansion of $IIX, IIY_l^{(j)}$
$\mathrm{E}X, \mathrm{E}Y_l^{(j)}$	The vector obtained by encryption of $IIIX, IIIY_l^{(j)}$
$x_i, \mathrm{I}x_i, \mathrm{II}x_i, \mathrm{II}x_i, ex_i$	The components of $X, IX, IIX, IIIX, EX$
$y_{li}^{(j)}, \mathrm{I}y_{li}^{(j)}, \mathrm{II}y_{li}^{(j)}, \mathrm{III}y_{li}^{(j)}, ey_{li}^{(j)}$	The components of $Y_l^{(j)}, IY_l^{(j)}, IIY_l^{(j)}, IIIY_l^{(j)}, EY_l^{(j)}$
p, a	Two large prime numbers chosen by the hospital
$\beta, \gamma, b_i, b_i^{(j)}$	Random numbers chosen by the hospital
k_l	The number of nearest neighbors selected by relief-kMW classifier for disease l
RW	Relief-Wasserstein distance between two pieces of data
θ	Intermediate parameters when calculating RW
$\mathrm{TS}_i, \mathrm{ID}_U$	Timestamp of the i-th moment and the user's ID
X_Q	Query request encrypted with PK_H by the user
Q	Query vector encrypted with PK_C by the hospital
R, R_l	Prediagnosis result and pre-diagnosis result encrypted with PK_H by the cloud
A_l	Medical advice given by the hospital
RA_l	Prediagnosis result and medical advice encrypted with PK_U by the hospital

require decryption to obtain the original data and also provides relatively accurate prediagnosis services.

Theorem 3. *Assuming that RW_1 represents the Relief-Wasserstein distance between IIIX and $IIIY_l^{(j)}$, RW represents the Relief-Wasserstein distance between the two original data, then RW is equal to RW_1.*

Proof.The calculation process of the Relief-Wasserstein distance RW between the original data is as follows:

$$\theta_0 = 0,$$

$$\theta_1 = \theta_0 + w_1 * \left(x_1 - y_{l1}^{(j)}\right) = w_1x_1 - w_1y_{l1}^{(j)},$$

$$\theta_2 = \theta_1 + w_2 * \left(x_2 - y_{l2}^{(j)}\right) = (w_1x_1 + w_2x_2) - \left(w_1y_{l1}^{(j)} + w_2y_{l2}^{(j)}\right), \cdots,$$

$$\theta_{n_l} = \theta_{n_l-1} + w_{n_l} * \left(x_{n_l} - y_{ln_l}^{(j)}\right) = \left(w_1x_1 + \cdots + w_{n_l}x_{n_l}\right) - \left(w_1y_{l1}^{(j)} + \cdots + w_{n_l}y_{ln_l}^{(j)}\right),$$

$$RW = \sum_{i=1}^{n_l}|\theta_i| = |\theta_1| + |\theta_2| + \cdots + \left|\theta_{n_l}\right| = \left|w_1x_1 - w_1y_{l1}^{(j)}\right| + \left|(w_1x_1 + w_2x_2) - \left(w_1y_{l1}^{(j)} + w_2y_{l2}^{(j)}\right)\right| + \cdots + \left|\left(w_1x_1 + \cdots + w_{n_l}x_{n_l}\right) - \left(w_1y_{l1}^{(j)} + \cdots + w_{n_l}y_{ln_l}^{(j)}\right)\right|. \tag{12}$$

RW_1 represents the Relief-Wasserstein distance between IIIX and $IIIY_l^{(j)}$, and then the calculation process of RW_1 is

```
Input: disease $l$, medical vector $Y_l^{(j)} = (y_{l1}^{(j)}, y_{l2}^{(j)}, \cdots, y_{ln_l}^{(j)}), j \in [1, N_l]$ (or $X = (x_1, x_2, \cdots, x_{n_l})$), feature weight vector $\text{weight}_l = (w_{l1}, w_{l2}, \cdots, w_{ln_l})$.
Output: the preprocessed vector $EY_l^{(j)} = (ey_{l1}^{(j)}, ey_{l2}^{(j)}, \cdots, ey_{ln_l+2}^{(j)}), j \in [1, N_l]$ (or $EX = (ex_1, ex_2, \cdots, ex_{n_l+2})$).
Generate two large prime numbers $p, a$ and a random number $\beta$ (or $\gamma$) and satisfy $\text{len}(p) = \text{len}(t_1)$, $\text{len}(a) = \text{len}(t_2)$ and $\beta \in Z_p$ (or $\gamma \in Z_p$);
The feature weight is assigned to get $IY_l^{(j)} = (Iy_{l1}^{(j)}, Iy_{l2}^{(j)}, \cdots, Iy_{ln_l}^{(j)}) = (w_{l1}y_{l1}^{(j)}, w_{l2}y_{l2}^{(j)}, \cdots, w_{ln_l}y_{ln_l}^{(j)})$, $j \in [1, N_l]$ (or $IX = (Ix_1, Ix_2, \cdots, Ix_{n_l}) = (w_{l1}x_{l1}, w_{l2}x_{l2}, \cdots, w_{ln_l}x_{ln_l})$);
The vector value is iteratively transformed to get $IIY_l^{(j)} = (IIy_{l1}^{(j)}, IIy_{l2}^{(j)}, \cdots, IIy_{ln_l}^{(j)}) = (w_{l1}y_{l1}^{(j)}, w_{l1}y_{l1}^{(j)} + w_{l2}y_{l2}^{(j)}, \cdots, w_{l1}y_{l1}^{(j)} + w_{l2}y_{l2}^{(j)} + \cdots + w_{ln_l}y_{ln_l}^{(j)}), j \in [1, N_l]$ (or $IIX = (IIx_1, IIx_2, \cdots, IIx_{n_l}) = (w_{l1}x_{l1}, w_{l1}x_{l1} + w_{l2}x_{l2}, \cdots, w_{l1}x_{l1} + w_{l2}x_{l2} + \cdots + w_{ln_l}x_{ln_l})$);
The vector is expanded forward to get $IIIY_l^{(j)} = (IIIy_{l1}^{(j)}, IIIy_{l2}^{(j)}, \cdots, IIIy_{ln_l+2}^{(j)}) = (0, 0, w_{l1}y_{l1}^{(j)}, w_{l1}y_{l1}^{(j)} + w_{l2}y_{l2}^{(j)}, \cdots, w_{l1}y_{l1}^{(j)} + w_{l2}y_{l2}^{(j)} + \cdots + w_{ln_l}y_{ln_l}^{(j)}), j \in [1, N_l]$ (or $IIIX = (IIIx_1, IIIx_2, \cdots, IIIx_{n_l+2}) = (0, 0, w_{l1}x_{l1}, w_{l1}x_{l1} + w_{l2}x_{l2}, \cdots, w_{l1}\text{x}_{l1} + w_{l2}x_{l2} + \cdots + w_{ln_l}x_{ln_l})$);
for $i = 1$ to $n_l + 2$ do
    Generate a random number $b_i^{(j)}$ (or $b_i$) and satisfy $\text{len}(b_i^{(j)} (\text{or } b_i)) = \text{len}(t_3)$;
    if $\text{III}y_{li}^{(j)} (\text{or III}x_i) \neq 0$ do
        $ey_{li}^{(j)} = \beta(a \cdot \text{III}y_{li}^{(j)} + b_i^{(j)}) \bmod p$ (or $ex_i = \gamma(a \cdot IIIx_i + b_i) \bmod p$);
    else do
        $ey_{li}^{(j)} = \beta \cdot b_i^{(j)} \bmod p$ (or $ex_i = \gamma \cdot b_i \bmod p$);
    end if
end for
return $EY_l^{(j)}, j \in [1, N_l]$ (or $EX$).
```

Algorithm 2: Preprocessing for medical instances.

```
Input: the preprocessed query vector $EX$, $\beta$, $\gamma$, $l$ and $k_l$.
Output: pre-diagnosis result $R$.
$EX = \gamma^{-1} \cdot EX$;
for $j = 1$ to $N_l$ do
    set $RW = 0$;
    for $i = 1$ to $n_l + 2$ do
        $RW = RW + |ex_i - \beta^{-1} \cdot ey_{li}^{(j)}|$;
    end for
end for
Select the $k_l$ data with the smallest $RW$ between the $N_l$ data and the query vector, and use most of the classification results in the $k_l$ data as the pre-diagnostic result $R$;
return $R$.
```

Algorithm 3: Prediagnosis service algorithm of the Relief-kMW classifier.

as follows:

$$
\begin{aligned}
\theta_1 &= IIIx_1 - IIIy_{l1}^{(j)} = 0 - 0 = 0,\\
\theta_2 &= IIIx_2 - IIIy_{l2}^{(j)} = 0 - 0 = 0,\\
\theta_3 &= IIIx_3 - IIIy_{l3}^{(j)} = w_1x_1 - w_1y_{l1}^{(j)},\\
\theta_4 &= IIIx_4 - IIIy_{l4}^{(j)} = (w_1x_1 + w_2x_2) - \left(w_1y_{l1}^{(j)} + w_2y_{l2}^{(j)}\right), \cdots,\\
\theta_{n_l+2} &= IIIx_{n_l+2} - IIIy_{ln_l+2}^{(j)} = \left(w_1x_1 + w_2x_2 + \cdots + w_{n_l}x_{n_l}\right) - \left(w_1y_{l1}^{(j)} + w_2y_{l2}^{(j)} + \cdots + w_{n_l}y_{ln_l}^{(j)}\right),\\
RW_1 &= \sum_{i=1}^{n_l+2} |\theta_i| = |\theta_1| + |\theta_2| + \cdots + \left|\theta_{n_l+2}\right| = 0 + 0 + \left|w_1x_1 - w_1y_{l1}^{(j)}\right|\\
&\quad + \left|(w_1x_1 + w_2x_2) - \left(w_1y_{l1}^{(j)} + w_2y_{l2}^{(j)}\right)\right| + \cdots + \left|(w_1x_1 + \cdots + w_{n_l}x_{n_l})\right.\\
&\quad \left. - \left(w_1y_{l1}^{(j)} + \cdots + w_{n_l}y_{ln_l}^{(j)}\right)\right| = RW.
\end{aligned}
\tag{13}
$$

Table 2: Selection of the optimal k value.

k	Accuracy (%)	Computation time (s)
$k_{\mathrm{WBC}} = 5$	97.87	0.3359 ± 0.0197
$k_{\mathrm{WBC}} = 7$	98.99	0.3438 ± 0.0156
$k_{\mathrm{MM}} = 9$	99.50	0.3672 ± 0.0134
$k_{\mathrm{WBC}} = 11$	99.50	0.3828 ± 0.0245
$k_{\mathrm{WBC}} = 13$	99.50	0.3984 ± 0.0234
$k_{\mathrm{WBC}} = 15$	99.50	0.4063 ± 0.0313
$k_{\mathrm{MM}} = 5$	94.04	0.6094 ± 0.1094
$k_{\mathrm{MM}} = 7$	97.07	0.6641 ± 0.1328
$k_{\mathrm{MM}} = 9$	97.09	0.6953 ± 0.1328
$k_{\mathrm{MM}} = 11$	97.78	0.6718 ± 0.0938
$k_{\mathrm{MM}} = 13$	96.52	0.7109 ± 0.1016
$k_{\mathrm{MM}} = 15$	96.50	0.7734 ± 0.1484

In summary, by Theorem 3, we can sssget that RW_1 is equal to RW.

Theorem 4. *Assuming that RW_1 represents the Relief-Wasserstein distance between IIIX and $IIIY_l^{(j)}$, RW_2 represents the Relief-Wasserstein distance between EX and $EY_l^{(j)}$, and then RW_1 and RW_2 are approximately in a fixed multiple relationship.*

Proof.RW_1 represents the Relief-Wasserstein distance between IIIX and $IIIY_l^{(j)}$, and then the calculation process of RW_1 is as follows:

$$\theta_1 = IIIx_1 - IIIy_{l1}^{(j)},$$

$$\theta_2 = IIIx_2 - IIIy_{l2}^{(j)},$$

$$\theta_4 = IIIx_4 - IIIy_{l4}^{(j)},$$

$$\cdots,$$

$$\theta_{n_l+2} = IIIx_{n_l+2} - IIIy_{ln_l+2}^{(j)},$$

$$RW_1 = \sum_{i=1}^{n_l+2} |\theta_i| = |\theta_1| + |\theta_2| + \cdots + |\theta_{n_l+2}| = \left| IIIx_1 - IIIy_{l1}^{(j)} \right| + |IIIx_2 - IIIy_{l2}^{(j)}\Big| + \cdots + \left| IIIx_{n_l+2} - IIIy_{ln_l+2}^{(j)} \right|. \tag{14}$$

RW_2 represents the Relief-Wasserstein distance between EX and $EY_l^{(j)}$, and then the calculation of RW_2 is as follows:

$$\theta_1 = \gamma^{-1} \cdot ex_1 - \beta^{-1} \cdot ey_{l1}^{(j)} = \gamma^{-1} \cdot \gamma(a \cdot IIIx_1 + b_1) \bmod p - \beta^{-1} \cdot \beta\left(a \cdot IIIy_{l1}^{(j)} + b_1^{(j)}\right) \bmod p = a \cdot \left(IIIx_1 - IIIy_{l1}^{(j)}\right) + b_1 - b_1^{(j)} \bmod p,$$

$$\theta_2 = \gamma^{-1} \cdot ex_2 - \beta^{-1} \cdot ey_{l2}^{(j)} = \gamma^{-1} \cdot \gamma(a \cdot IIIx_2 + b_2) \bmod p - \beta^{-1} \cdot \beta\left(a \cdot IIIy_{l2}^{(j)} + b_2^{(j)}\right) \bmod p = a \cdot \left(IIIx_2 - IIIy_{l2}^{(j)}\right) + b_2 - b_2^{(j)} \bmod p,$$

$$\cdots,$$

$$\theta_{n_l+2} = \gamma^{-1} \cdot ex_{n_l+2} - \beta^{-1} \cdot ey_{ln_l+2}^{(j)} = \gamma^{-1} \cdot \gamma\left(a \cdot IIIx_{n_l+2} + b_{n_l+2}\right) \bmod p - \beta^{-1} \cdot \beta\left(a \cdot IIIy_{ln_l+2}^{(j)} + b_{n_l+2}^{(j)}\right) \bmod p = a \cdot \left(IIIx_{n_l+2} - IIIy_{ln_l+2}^{(j)}\right) + b_{n_l+2} - b_{n_l+2}^{(j)} \bmod p,$$

$$\begin{aligned} RW_2 = \sum_{i=1}^{n_l+2} |\theta_i| &= |\theta_1| + |\theta_2| + \cdots + |\theta_{n_l+2}| \\ &= \left| a \cdot \left(IIIx_1 - IIIy_{l1}^{(j)}\right) + b_1 - b_1^{(j)} \bmod p \right| \\ &\quad + \left| a \cdot \left(IIIx_2 - IIIy_{l2}^{(j)}\right) + b_2 - b_2^{(j)} \bmod p \right| + \cdots \\ &\quad + \left| a \cdot \left(IIIx_{n_l+2} - IIIy_{ln_l+2}^{(j)}\right) + b_{n_l+2} - b_{n_l+2}^{(j)} \bmod p \right|. \end{aligned} \tag{15}$$

Available from $t_1 > t_2 > t_3$, we can get the followings:

$$\begin{aligned} RW_2 = &\left| a \cdot \left(IIIx_1 - IIIy_{l1}^{(j)}\right) + b_1 - b_1^{(j)} \right| \\ &+ \left| a \cdot \left(IIIx_2 - IIIy_{l2}^{(j)}\right) + b_2 - b_2^{(j)} \right| + \cdots \\ &+ \left| a \cdot \left(IIIx_{n_l+2} - IIIy_{ln_l+2}^{(j)}\right) + b_{n_l+2} - b_{n_l+2}^{(j)} \right| \end{aligned} \tag{16}$$

From $t_1 > t_2 > t_3$, we can get that $b_i - b_i^{(j)}$ is negligible relative to the large prime number a, and then

$$\begin{aligned} \mathrm{RW}_2 &\approx \left| a \cdot \left(\mathrm{III}x_1 - \mathrm{III}y_{l1}^{(j)}\right) \right| + \left| a \cdot \left(\mathrm{III}x_2 - \mathrm{III}y_{l2}^{(j)}\right) \right| + \cdots \\ &\quad + \left| a \cdot \left(\mathrm{III}x_{n_l+2} - \mathrm{III}y_{ln_l+2}^{(j)}\right) \right| \\ &= a \cdot \left(\left| \mathrm{III}x_1 - \mathrm{III}y_{l1}^{(j)} \right| + \left| \mathrm{III}x_2 - \mathrm{III}y_{l2}^{(j)} \right| + \cdots + \left| \mathrm{III}x_{n_l+2} - \mathrm{III}y_{ln_l+2}^{(j)} \right| \right) \\ &= a \cdot \mathrm{RW}_1. \end{aligned} \tag{17}$$

Finally, Theorem 4 holds. And we can get that RW_1 and RW_2 are approximately in a fixed multiple relationship. It can be obtained from Theorems 3 and 4 that the cloud can still obtain accurate pre-diagnosis results through Algorithm 3 without obtaining the original data.

Table 3: Comparison of accuracy.

Accuracy		OMPD	OMPD-*k*MW	EPDK	PPDP
WBC	Malignant (100)	99 (99.0%)	97 (97.0%)	93 (93.0%)	85 (85.0%)
	Benign (100)	97 (97.0%)	95 (95.0%)	94 (94.0%)	81 (81.0%)
	Overall (200)	196 (98.0%)	192 (96.0%)	187 (93.5%)	166 (83.0%)
MM	Malignant (100)	97 (97.0%)	97 (97.0%)	95 (95.0%)	83 (83.0%)
	Benign (100)	98 (98.0%)	96 (96.0%)	98 (98.0%)	86 (86.0%)
	Overall (200)	195 (97.5%)	193 (96.5%)	193 (96.5%)	169 (84.5%)

5.2. Security Analysis. The focus of security analysis is whether OMPD can protect the privacy of users' medical data, the privacy of medical instances in hospitals, and the confidentiality of cloud diagnostic models.

The user's query vector $X = (x_1, x_2, \cdots, x_{n_l})$ and the medical instance $Y_l^{(j)} = (y_{l1}^{(j)}, y_{l2}^{(j)}, \cdots, y_{ln_l}^{(j)})$, $j \in [1, N_l]$ in the hospital data subset D_l are privacy-preserving. In the initialization stage, in order to prevent the privacy of medical instances from leaking, the hospital expands two dimensions with 0 for each medical instance after feature weight distribution and vector value iteration, which can prevent the cloud and illegal users from obtaining real medical instances. Each $b_{li}^{(j)}$ in the encryption calculation is randomly generated; so, III$y_{li}^{(j)}$ is protected. When p, a, $b_{li}^{(j)}$, and weight$_l$ are unknown, the cloud cannot obtain the original medical data. And the random number $b_{li}^{(j)}$ generated is independent of each other at each time. Only the hospital knows the data processing rules, and the cloud cannot infer the original medical data. Therefore, the medical instance set $Dataset = \{D_l = \{Y_l^{(j)}, j \in [1, N_l]\}, l \in [1, L]\}$ is kept secret during the calculation process. Similarly, the query vector of the user $X = (x_1, x_2, \cdots, x_{n_l})$ is also kept secret.

The Relief-*k*MW classifier is confidential. In the operating calculation phase, each medical instance $Y_l^{(j)} = (y_{l1}^{(j)}, y_{l2}^{(j)}, \cdots, y_{ln_l}^{(j)})$, $j \in [1, N_l]$ and query vector $X = (x_1, x_2, \cdots, x_{n_l})$ are preprocessed by the hospital before sending to the cloud. For two same query vectors, they should have the same Relief-Wasserstein distance between the same medical instances, but the random number $b_{li}^{(j)}$ and b_i generated by each data processing are different and the Relief-Wasserstein distance calculated by the query vector is also different, which ensures that even the same user cannot obtain medical instance information after multiple queries. And each RW calculated by the Relief-*k*MW classifier is confidential, and the cloud uses the hospital's public key PK$_H$ to encrypt the query result. The hospital uses the user's public key PK$_U$ to encrypt the query result and the advice; so, only the corresponding user can decrypt the query result. Moreover, the user and the cloud cannot communicate directly during the query. Although the cloud knows the final prediction result, it cannot obtain the corresponding user's information. And the user cannot obtain the detailed information of the Relief-*k*MW classifier in the cloud. Therefore, the Relief-*k*MW classifier is confidential.

OMPD ensures communication security. Our scheme uses BLS signature to protect the information of each interaction. The signature is proved to be safe under the Diffie-Hellman problem of the random prediction model [36]. In addition, any illegal user cannot successfully submit a query request to the hospital because there is no key. Signature authentication can ensure that the message is not maliciously tampered with during transmission. Even if someone maliciously intercepts the message, the effective information in the message cannot be obtained because there is no key.

In summary, our OMPD can provide privacy-preserving medical prediagnosis services.

6. Performance Evaluation

In this section, multiple experiments are conducted to verify the accuracy and efficiency of OMPD from multiple dimensions of accuracy, computation overhead, and communication overhead.

6.1. Experiment Configuration. We implement OMPD with Python programming language and evaluate the computation and communication overhead of OMPD. We carry out our experiments on the device with CPU Intel(R) Xeon(R) Silver 4114, 2.20GHz, and Memory 32G. We choose two real data sets: Wisconsin Breast Cancer (WBC) and Mammographic Mass (MM) in the UCI machine learning library to evaluate the accuracy of the OMPD. In the comparative experiment, OMPD uses the Relief-*k*MW classifier, while OMPD-*k*MW uses the *k*MW classifier, EPDK [33] is a two-party interaction pre-diagnosis program using *k*NN classifiers, and PPDP [32] is a three-party interaction prediagnostic program using a single-layer perceptron trained with encryption matrices as the classifier.

The WBC contains 683 instances, including 444 benign instances and 239 malignant instances. Each instance contains 9 attributes. The MM contains 830 instances, of which 427 instances are benign and 403 instances are malignant. Each instance contains 5 attributes.

6.2. Selection of the Optimal Value of k. For WBC and MM, we randomly selected 100 malignant instances and 100 benign instances as the test data set to evaluate the accuracy of OMPD. And the rest were used as the training data set. Then, we performed 100 calculations to compare the average accuracy and computation time under different values of *k*. As shown in Table 2, when $k_{\text{WBC}} = 9$ and $k_{\text{MM}} = 11$, the classification accuracy of OMPD is the highest, and the

Table 4: Comparison of computation complexity.

Complexity	OMPD	OMPD-*k*MW	EPDK	PPDP
User	—	—	$C_m \cdot (N_l + 3n_l + 2) + C_a \cdot (N_l + 2n_l - 1)$	—
Hospital	$C_m \cdot (3n_l + 2) + C_a \cdot (2n_l - 1)$	$C_m \cdot (2n_l + 2) + C_a \cdot (2n_l - 1)$	—	$C_m \cdot \left(2n_l^3 + n_l^2 + n_l\right) + C_a \cdot \left(2n_l^3 - 2n_l^2 + n_l - 1\right)$
Cloud/server	$C_m \cdot (N_l n_l + 2N_l + n_l + 2) + C_a \cdot (2N_l n_l + 4N_l)$	$C_m \cdot (N_l n_l + 2N_l + n_l + 2) + C_a \cdot (2N_l n_l + 4N_l)$	$C_m \cdot (4N_l n_l + 4N_l) + C_a \cdot (N_l n_l + N_l + n_l + 1)$	$C_m \cdot 2n_l^3 + C_a \cdot \left(2n_l^3 - 2n_l^2 + n_l - 1\right)$
Total	$C_m \cdot (N_l n_l + 2N_l + 4n_l + 4) + C_a \cdot (2N_l n_l + 4N_l + 2n_l - 1)$	$C_m \cdot (N_l n_l + 2N_l + 3n_l + 4) + C_a \cdot (2N_l n_l + 4N_l + 2n_l - 1)$	$C_m \cdot (4N_l n_l + 5N_l + 3n_l + 2) + C_a \cdot (N_l n_l + 2N_l + 3n_l)$	$C_m \cdot \left(4n_l^3 + n_l^2 + n_l\right) + C_a \cdot \left(4n_l^3 - 4n_l^2 + 2n_l - 2\right)$

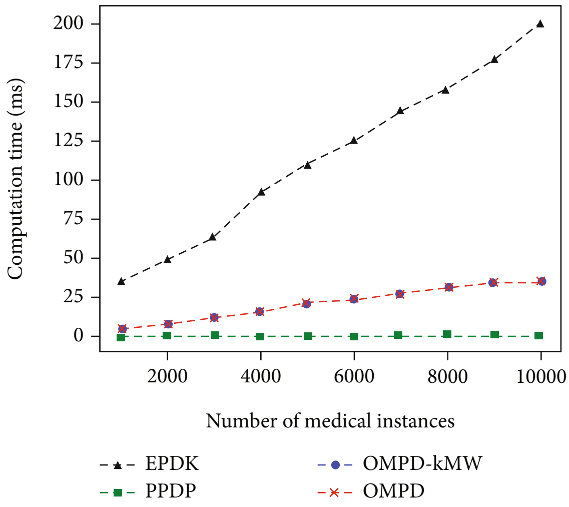

Figure 2: Comparison of computation overhead.

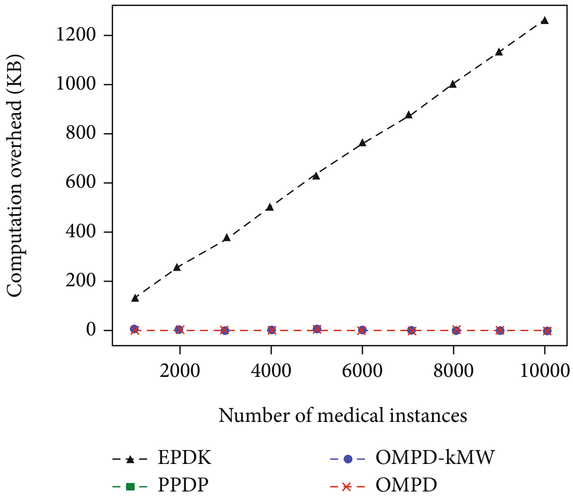

Figure 3: Comparison of communication overhead.

calculation time-consuming swing is the smallest. Therefore, in the following experiment, the values of k_{WBC} and k_{MM} are 9 and 11, respectively.

6.3. Evaluation of Accuracy. In order to verify the accuracy of our method, we conducted an accuracy comparison experiment, as shown in Table 3. The results show that the classification accuracy of OMPD is significantly higher than the other three schemes. This means that our OMPD still has a high classification accuracy after the data is preprocessed by feature weight distribution, iteration, expansion, and encryption. And it can provide high-precision online medical prediagnosis services for medical users. Please note that the accuracy of OMPD is higher than OMPD-*k*MW, which means that the feature weights obtained by OMPD using the Relief feature weight distribution algorithm, which increases the impact of features that have a greater contribution to the classification and reduces the impact of features with low contributions. The experimental results verify that the accuracy of using Relief-*k*MW is higher than that of *k* MW.

6.4. Evaluation of Computational Efficiency. In order to verify that OMPD can provide efficient online medical prediagnosis services for medical users, this section evaluates the efficiency of computation. During the system operation, when the user generates the query vector and sends it to the hospital, the hospital needs to preprocess the query vector through $(3n_l + 2)$ multiplication (division) and $(2n_l - 1)$ addition (subtraction) operation. After the cloud receives the query request, it needs to go through $(N_l n_l + 2N_l + n_l + 2)$ multiplication (division) and $(2N_l n_l + 4N_l)$ addition (subtraction) operation to calculate the diagnosis result. Let C_a and C_m denote the running time of addition and multiplication, respectively. Then, the overall computation complexity of OMPD is $C_m \cdot (N_l n_l + 2N_l + 4n_l + 4) + C_a \cdot (2N_l n_l + 4N_l + 2n_l - 1)$ during system operation. As shown in Table 4, it can be seen that OMPD-*k*MW only lacks a multiplication operation (the calculation of multiplying feature weights and feature values) than our OMPD. The computational complexity of PPDP in the operating phase of the system has nothing to do with the amount of data N_l. Its computational complexity is concentrated in the system initialization stage, and a lot of calculations are required to continuously update the weights to converge. The multiplication (division) in OMPD is far less than that in EPDK, while C_m is more than C_a. Therefore, the computational complexity of OMPD is lower than that of EPDK, and the gap becomes more obvious with the increase of data.

However, we cannot intuitively see the comparison of the computational time from Table 4. So, we conducted a comparison experiment of computational overhead. As shown in Figure 2, as the number of medical instances increases, the computation time of EPDK has grown rapidly, and OMPD and OMPD-*k*MW have grown slowly, while PPDP has remained almost unchanged. This is because EPDK takes more time to perform operations such as multiplication (division). In addition, a large number of medical data stored by EPDK providers are collected directly from various medical service sites without encryption and other privacy protection measures, and there is a serious risk of privacy leakage. Our OMPD and OMPD-*k*MW, as shown in Table 4, require significantly less calculations than EPDK. However, PPDP only needs to calculate the query data and the weight to get the diagnosis result; so, the computation time of PPDP in the running phase remains unchanged, but PPDP generates a lot of computation and communication overhead in the initialization phase to update the weights to converge. And PPDP ignores the protection of the communication process. Users send sensitive information directly to the hospital, which is vulnerable to attacks and theft by malicious third parties.

6.5. Evaluation of Communication Overhead. The communication overhead comparison of the four schemes in the system operation phase is shown in Figure 3. Assuming that the medical advice contains 100 words, each word contains an average of 5 characters, plus punctuation and spaces, one word approximately contains 6 characters, and each character occupies 1 byte. The total communication overhead of OMPD is about 0.735 KB (including 600B medical advice). As the number of medical instances increases, the total communication overhead of OMPD, OMPD-*k*MW, and PPDP remains unchanged, while EPDK is much higher than the other three schemes. Because these three schemes obtain the diagnosis results directly in the cloud, only the results need to be sent during the communication process. The server in the EPDK only calculates the two intermediate values of the Spearman distance between the query vector and N_l medical instances in database. The server needs to send $2N_l$ intermediate values to the user, and the user finally calculates the diagnosis result. Therefore, the larger the N_l, the greater the communication overhead of EPDK. And users cannot get corresponding medical advice in time.

In summary, OMPD can balance accuracy and efficiency to achieve high-precision online medical prediagnosis with lower computational complexity and communication overhead. At the same time, it also provides timely professional medical advice.

7. Conclusions

In this article, we propose an efficient online multiparty interactive medical prediagnosis scheme with privacy protection, called OMPD, which can protect the privacy with low computation and communication overhead. Accuracy analysis showed that our OMPD can obtain more accurate classification results without decryption. The security analysis demonstrated its security strength and privacy protection capabilities. And its effectiveness was verified through comparative experiments. Future work is to further improve the classification accuracy and operating efficiency of the program while ensuring the strength and security of privacy protection.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Key Research and Development Program of Anhui, China under Grant No. 201904a05020071, the Program for Synergy Innovation in the Anhui Higher Education Institutions of China (Grant Nos. GXXT-2019-025 and GXXT-2020-012), in part by the Natural Science Research Project of Anhui Province (Grant No. 2108085MF218), in part by the University Natural Science Research Project of Anhui Province (No. KJ2020A0249), and in part by the Open Fund of Key Laboratory of Anhui Higher Education Institutes under Grant CS2020-006.

References

[1] 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.

[2] S. Mao, L. Zhang, and Z. Guan, “An LSTM&Topic-CNN model for classification of online Chinese medical questions,” *IEEE Access*, vol. 9, pp. 52580–52589, 2021.

[3] H. Zhu, X. Liu, R. Lu, and H. Li, “Efficient and privacy-preserving online medical prediagnosis framework using nonlinear SVM,” *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 3, pp. 838–850, 2016.

[4] X. Liu, H. Zhu, R. Lu, and H. Li, “Efficient privacy-preserving online medical primary diagnosis scheme on naive bayesian classification,” *Peer-to-Peer Networking and Applications*, vol. 11, no. 2, pp. 334–347, 2018.

[5] H. O. Sonkurt, A. Altnöz, E. Çimen, F. Kösger, and G. Öztürk, “The role of cognitive functions in the diagnosis of bipolar disorder: a machine learning model,” *International Journal of Medical Informatics*, vol. 145, article 104311, 2021.

[6] L. Zhang, Y. Huo, Q. Ge, Y. Ma, Q. Liu, and W. Ouyang, “A privacy protection scheme for IoT big data based on time and frequency limitation,” *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 5545648, 2021.

[7] Z. Zhou, Y. Tian, and C. Peng, “Privacy-preserving federated learning framework with general aggregation and multiparty entity matching,” *Wireless Communications and Mobile Computing*, vol. 2021, 2021.

[8] H. Li, Y. Wang, F. Guo, J. Wang, B. Wang, and C. Wu, “Differential privacy location protection method based on the Markov model,” *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 4696455, 2021.

[9] L. Sweeney, “K-anonymity: a model for protecting privacy,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[10] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, “L-diversity: privacy beyond k-anonymity,” *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, p. 3, 2007.

[11] C. Dwork, “Differential privacy: a survey of results,” in *International conference on theory and applications of models of computation*, pp. 1–19, Springer, Berlin, Heidelberg, 2008.

[12] A. Cheu, A. Smith, and J. Ullman, “Manipulation attacks in local differential privacy,” in *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 883–900, San Francisco, CA, USA, 2021.

[13] Q. Ye, H. Hu, X. Meng, and H. Zheng, “PrivKV: key-value data collection with local differential privacy,” in *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 317–331, San Francisco, CA, USA, 2019.

[14] R. L. Rivest, L. Adleman, and M. L. Dertouzos, “On data banks and privacy homomorphisms,” *Foundations of Secure Computation*, vol. 4, pp. 169–180, 1978.

[15] P. Paillier, “Public-key cryptosystems based on composite degree residuosity classes,” in *International conference on the theory and applications of cryptographic techniques*, pp. 223–238, Springer, Berlin, Heidelberg, 1999.

[16] D. Boneh, B. Lynn, and H. Shacham, “Short signatures from the Weil pairing,” in *International conference on the theory*

and application of cryptology and information security, pp. 514–532, Springer, Berlin, Heidelberg, 2001.

[17] H. Lee, D. Kang, Y. Lee, and D. Won, "Secure three-factor anonymous user authentication scheme for cloud computing environment," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 2098530, 2021.

[18] S. Wagh, D. Gupta, and N. Chandran, "SecureNN: 3-party secure computation for neural network training," *Proceedings on Privacy Enhancing Technologies*, vol. 2019, no. 3, pp. 26–49, 2019.

[19] Y. Zhang, M. Blanton, and G. Almashaqbeh, "Secure distributed genome analysis for GWAS and sequence comparison computation," *BMC Medical Informatics and Decision Making*, vol. 15, no. 5, pp. 1–12, 2015.

[20] G. Mathew and Z. Obradovic, "A privacy-preserving framework for distributed clinical decision support," in *2011 IEEE 1st International Conference on Computational Advances in Bio and Medical Sciences (ICCABS)*, pp. 129–134, Orlando, FL, USA, 2011.

[21] Y. Rahulamathavan, S. Veluru, R. C. W. Phan, J. A. Chambers, and M. Rajarajan, "Privacy-preserving clinical decision support system using gaussian kernel-based classification," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 1, pp. 56–66, 2013.

[22] R. Bost, R. A. Popa, S. Tu, and S. Goldwasser, "Machine learning classification over encrypted data," in *22nd Annual Network and Distributed System Security Symposium (NDSS)*, p. 4325, San Diego, California, USA, 2015.

[23] M. Barni, P. Failla, R. Lazzeretti, A. R. Sadeghi, and T. Schneider, "Privacy-preserving ECG classification with branching programs and neural networks," *IEEE Transactions on Information Forensics and Security*, vol. 53, no. 3, pp. 1020–1022, 2011.

[24] X. Liu, R. Lu, J. Ma, L. Chen, and B. Qin, "Privacy-preserving patient-centric clinical decision support system on naive Bayesian classification," *IEEE Journal of Biomedical and Health Informatics*, vol. 20, no. 2, pp. 655–668, 2015.

[25] J. Hua, G. Shi, H. Zhu, F. Wang, X. Liu, and H. Li, "AMPS: efficient and privacy-preserving medical primary diagnosis over outsourced cloud," *Information Sciences*, vol. 527, pp. 560–575, 2020.

[26] Y. Zhu, Z. Wang, and J. Wang, "Collusion-resisting secure nearest neighbor query over encrypted data in cloud, revisited," in *2016 IEEE/ACM 24th International Symposium on Quality of Service (IWQoS)*, pp. 1–6, Beijing, 2016.

[27] D. J. Wu, T. Feng, M. Naehrig, and K. E. Lauter, "Privately evaluating decision trees and random forests," *Proceedings on Privacy Enhancing Technologies*, vol. 2016, no. 4, pp. 335–355, 2016.

[28] J. Yuan and S. Yu, "Efficient privacy-preserving biometric identification in cloud computing," in *2013 Proceedings IEEE INFOCOM*, pp. 2652–2660, Turin, Italy, 2013.

[29] Q. Zhang, L. T. Yang, and Z. Chen, "Privacy preserving deep computation model on cloud for big data feature learning," *IEEE Transactions on Computers*, vol. 65, no. 5, pp. 1351–1362, 2015.

[30] S. Pan, S. Yan, and W. T. Zhu, "Security analysis on privacy-preserving cloud aided biometric identification schemes," in *Australasian Conference on Information Security and Privacy*, pp. 446–453, Melbourne, Australia, 2016.

[31] B. Wang, S. Yu, W. Lou, and Y. T. Hou, "Privacy-preserving multi-keyword fuzzy search over encrypted data in the cloud," in *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*, pp. 2112–2120, Toronto, ON, Canada, 2014.

[32] C. Zhang, L. Zhu, C. Xu, and R. Lu, "PPDP: an efficient and privacy-preserving disease prediction scheme in cloud-based e-healthcare system," *Future Generation Computer Systems*, vol. 79, pp. 16–25, 2018.

[33] D. Zhu, H. Zhu, X. Liu, H. Li, F. Wang, and H. Li, "Achieve efficient and privacy-preserving medical primary diagnosis based on kNN," in *2018 27th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–9, Hangzhou, China, 2018.

[34] K. Kira and L. A. Rendell, "The feature selection problem: traditional methods and a new algorithm," in *Proceedings of the Tenth National Conference on Artificial Intelligence*, pp. 129–134, San Jose, CA, 1992.

[35] X. Wu, "Top 10 algorithms in data mining," *Knowledge and Information Systems*, vol. 14, no. 1, pp. 1–37, 2007.

[36] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.