Future Generation Computer Systems 79 (2018) 16–25

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

PPDP: An efficient and privacy-preserving disease prediction scheme in cloud-based e-Healthcare system

Chuan Zhang [a], Liehuang Zhu [a], Chang Xu [a,*], Rongxing Lu [b]

[a] School of Computer Science & Technology, Beijing Institute of Technology, Beijing, China
[b] Faculty of Computer Science, University of New Brunswick, Fredericton, Canada

HIGHLIGHTS

- An efficient privacy-preserving disease prediction scheme (PPDP) is proposed.
- PPDP can train prediction models without leaking the privacy of sensitive data.
- Security analysis indicates that PPDP is secure under a well-defined threat model.
- The performance evaluation on different datasets demonstrates PPDP's efficiency.

ARTICLE INFO

Article history:
Received 18 October 2016
Received in revised form 23 June 2017
Accepted 3 September 2017
Available online 7 September 2017

Keywords:
Disease prediction
Privacy-preserving
Single-Layer Perceptron
Cloud computing

ABSTRACT

Disease prediction systems have played an important role in people's life, since predicting the risk of diseases is essential for people to lead a healthy life. The recent proliferation of data mining techniques has given rise to disease prediction systems. Specifically, with the vast amount of medical data generated every day, Single-Layer Perceptron can be utilized to obtain valuable information to construct a disease prediction system. Although the disease prediction system is quite promising, many challenges may limit it in practical use, including information security and prediction efficiency. In this paper, we propose an efficient and privacy-preserving disease prediction system, called PPDP. In PPDP, patients' historical medical data are encrypted and outsourced to the cloud server, which can be further utilized to train prediction models by using Single-Layer Perceptron learning algorithm in a privacy-preserving way. The risk of diseases for new coming medical data can be computed based on the prediction models. In particular, PPDP builds on new medical data encryption, disease learning and disease prediction algorithms that novelly utilize random matrices. Security analysis indicates that PPDP offers a required level of privacy protection. In addition, real experiments on different datasets show that computation costs of data encryption, disease learning and disease prediction are several magnitudes lower than existing disease prediction schemes.

1. Introduction

Medical institutions, extensively distributed in the global world to provide health services for patients, have to face a massive amount of electronic health data (EHR) today. According to the report released by consulting company EMC and research firm IDC, the global healthcare data have reached 153 exabytes (10^{18} bytes) in 2013, and would soon reach 2314 exabytes by 2020 [1]. These medical data, on the one hand, require a great amount of space for storage and management [2], on the other hand, might become meaningless if no appropriate techniques can be developed to find great potential values from them. Over the past two decades, data mining techniques have imposed a major impact on human's lifestyle by predicting human's behaviors and future trends [3]. These techniques are well appropriate to convert stored data into valuable information to provide decision support in the healthcare system, e.g., to improve diagnosis accuracy and speed up diagnosis time. Disease Prediction Systems (DPSs), with various of data mining techniques being applied, have drawn a great attention recently [4–14]. Single-Layer Perceptron (SLP) classifier, as one of the most popular data mining tools, has been widely used to predict a variety of diseases [15]. Despite its simplicity, it is more efficient and appropriate than some sophisticated techniques such as Support Vector Machine (SVM) [16], Naïve Bayesian classification [3] and so on.

The DPSs with SLP classifier have offered obvious advantages in healthcare system and opens a new way to predict patients'

* Corresponding author.
E-mail address: xuchang@bit.edu.cn (C. Xu).

http://dx.doi.org/10.1016/j.future.2017.09.002
0167-739X/

diseases. Nevertheless, its flourish still hinges on how to fully manage privacy issues and prediction efficiency, especially considering sensitive medical data are stored in an unauthorized third-party. Therefore, privacy-preserving data mining algorithms should be built for protecting the privacy of medical data. Prediction models, which are obtained by training the medical data and used to predict patients' diseases, are incapable of being exposed to the third party since they are considered as the hospital's own asset. Otherwise, the third party might abuse prediction models for disease diagnosis, which could damage hospitals' or other service providers' profit. Therefore, how to preserve the privacy of prediction models is also crucial for DPSs. Besides the privacy protection, prediction efficiency is another important factor which needs to be considered in designing a DPS. In particular, DPSs require learning prediction models from a large amount of medical data. Traditional cryptographic primitives such as Paillier homomorphic encryption, though can protect medical data, are not highly efficient and practical. To the best of our knowledge, there is no privacy-preserving SLP-based DPS proposed which not only offers the required privacy protection, but also is highly practical and efficient.

In this paper, to address the aforementioned problems, we propose an efficient and privacy-preserving DPS by using SLP learning algorithm, called PPDP. With PPDP, patients can get privacy-preserving disease prediction services. Besides, the cloud server learns no privacy of medical data and prediction models. If a patient is willing to submit his/her symptom information (e.g., blood pressure, heart rate, etc.) to a hospital for disease prediction. Then, the hospital will submit the encrypted symptoms to the cloud. The cloud will use the encrypted prediction models trained by it to diagnose the diseases without getting privacy information. Then, the hospital will return the prediction results to the patient.

Specifically, the contributions of this paper can be summarized as follows.

- First, this paper proposes a secure PPDP scheme which allows the cloud server to diagnose patients' diseases without leaking sensitive information. In PPDP, patients' historical medical data are securely stored in the cloud and can be used to build prediction models by using the SLP learning algorithm. Based on the prediction models, the cloud server can diagnose patients' diseases in a privacy-preserving way.
- Second, to improve efficiency and minimize privacy disclosure, PPDP builds on new and secure medical data encryption and prediction algorithms that novelly utilize random matrices to protect privacy and facilitate secure outsourced computation of ciphertexts. Even there exists a collusion between patients and the cloud server, no party can obtain sensitive information. By analyzing capabilities of adversaries in different attack scenarios, this paper indicates that PPDP is practically-secure and can achieve a required level of security.
- Third, to validate accuracy and efficiency of the PPDP, we conduct experiments on real and synthetic datasets. Extensive simulation demonstrates that PPDP can help to diagnose diseases with an acceptable success rate and is efficient at all data encryption, disease learning and disease prediction processes.

The remainder of this paper is organized as follows: Section 2 presents the problem formulation. In Section 3, we provide preliminaries, which serve as the basis of our scheme. The proposed PPDP scheme is described in Section 4, followed by the security analysis in Section 5 and the performance evaluation in Section 6. The related work is given in Section 7, and we conclude this work in Section 8.

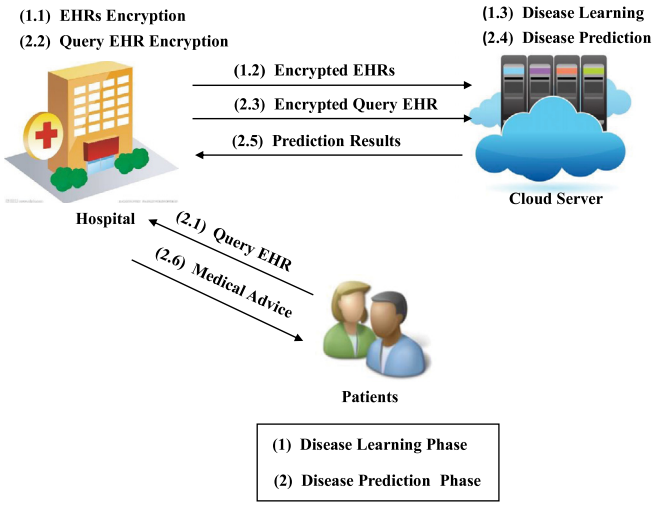


Fig. 1. Architecture of the privacy-preserving disease prediction cloud-based system.

2. Models and design goal

In this section, we formulate the system model, threat model and identify design goals.

2.1. System model

In this work, we mainly focus on how to securely train cloud-based SLP classifier and use this classifier to decide patients' diseases without leaking privacy information. In such a way, the system consists of three entities: a hospital, a cloud server and patients, as shown in Fig. 1.

- Hospital: The hospital is an indispensable entity, which is trusted by all patients, and in charge of generating secret keys and encrypting all medical data. It provides the medical data including patients' symptoms and confirmed diseases, which can be used to train the SLP classifier.
- Cloud server: The cloud server contains unlimited storage space, which is able to store all medical data in the system. Hospitals which have limited storage space can outsource their medical data to the cloud server. In addition, the cloud server has computation abilities to execute the calculations over the stored data including disease learning and disease prediction.
- Patients: Patients have some symptom information (e.g., blood pressure, heart rate, etc.), which can be obtained from doctors or collected by some sensors [17–19]. The symptoms are sent to the hospital for disease diagnosis.

2.2. Threat model

In our threat model, the cloud server is considered as *honest but curious* [20]. That is, it will honestly follow the designed protocol but attempt to disclose sensitive medical information as much as possible. In practical applications, adversaries have different levels of background knowledge and attack capabilities. Similar to [21], we characterize the attack-specific capabilities in three scenarios. The threat model is defined as follows:

- *Level-I*: The adversary can observe all encrypted EHRs and queries. *Level-I* attack follows the well-known ciphertext-only attack model [22].

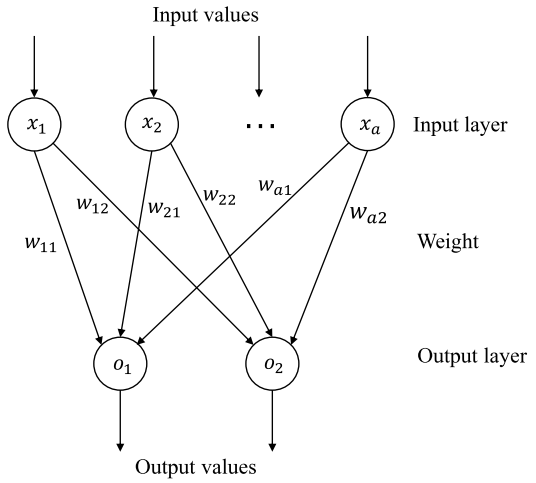


Fig. 2. Configuration of single-layer perceptron.

- *Level-II:* On the basis of *Level-I* attack, the adversary is assumed to know some samples of EHRs, but has no idea about the corresponding ciphertexts. This model follows the known-sample attack model [23].
- *Level-III:* On the basis of *Level-I* and *Level-II* attack, besides the encrypted EHRs database, the adversary knows the ciphertexts generated by patients for prediction. Moreover, the adversary can construct arbitrary EHRs of interest to execute disease prediction. *Level-III* attack is similar to the chosen-plaintexts attack model [22].

2.3. Design goals

Based on the aforementioned system model and threat model, the primary design goal of the proposed PPDP scheme is to develop an efficient and privacy-preserving disease prediction system. The following design goals should be guaranteed:

- *Learning and prediction:* With the proposed scheme, the cloud server can obtain a set of prediction models from the medical data. When patients submit query EHRs, PPDP can give prediction results based on the prediction models.
- *Privacy protection:* Sensitive EHRs and prediction models should be well protected. The cloud server and patients cannot learn anything other than what they have known.
- *Efficiency:* Hospitals always have limited computational resources. PPDP should efficiently train medical data using privacy-preserving SLP algorithm. When a patient submits query EHRs, PPDP should efficiently output the prediction results.

3. Preliminaries

In this section, we first review the SLP learning algorithm [24], which serves as the basis of the proposed PPDP scheme. Before that, we first list the notations which are used in Algorithm 1 and Algorithm 2 in Table 1.

SLP is one of the most popular neural network architectures which can decide whether an input belongs to one class or another [24]. Fig. 2 shows a configuration of SLP. Specifically, it contains two stages: input layer and output layer. Input nodes are denoted as a set of vectors $\{x_1, x_2, \ldots, x_a\}$ where $x_i \in \mathbb{R}^d$ and output layer nodes are denoted as $\{o_1, o_2\}$ which are belonging to $\{-1, 1\}$. w_{ij} is utilized to represent a weight connecting the input layer node x_i and the output layer node o_j, where $1 \leq i \leq a$, $1 \leq j \leq 2$.

For the SLP learning algorithm, each input node (i.e., vector x_i) has a corresponding output (i.e., desired value $o_i \in \{-1, 1\}$). The goal is to find a weight to generate expected outputs for all input layer nodes. A general SLP learning procedure is described by Algorithm 1. The weight is initialized as a d-dimension vector whose elements are set as small random values. In the input layer, SLP receives a set of input nodes. Then, it calculates the values at each layer with the weight and sign function (i.e., an active function to execute a two-class classification task onto the space $\{-1, 1\}$) according to Eq. (1), and further checks whether the result s_i is equal to the desired value.

$$\begin{aligned} s_i &= sign(f(w_k^T x_i) \\ &= sign(f(w_{k1} * x_{i1} + w_{k2} * x_{i2} + \cdots + w_{kd} * x_{id})). \end{aligned} \tag{1}$$

If $s_i \neq o_i$, the weight will be updated according to Eqs. (2) and (3) and the learning procedure is repeated. The learning process will be terminated if a convergence criterion (e.g., a threshold or a max number of iterations) is satisfied.

$$\triangle w_k = o_i x_i \tag{2}$$

$$w_{k+1} = w_k + \eta \triangle w_k. \tag{3}$$

Algorithm 1: Single-Layer Perceptron Learning Algorithm.

Input: d-dimension vectors $< x_i >_{i=1}^{a}$, learning rate η, desired value $< o_i >_{i=1}^{a} \in \{-1, 1\}$, $iteration_{max}$, sign function $sign(\cdot)$;
Output: Weight: w;

```
Randomly initialize w;
for iteration = 1, 2, ..., iteration_max do
    for i = 1, 2, ..., a do
        compute the sign function (see Eq. (1));
        if s_i ≠ o_i then
            update the weight (see Eqs. (2), (3));
        else
            //Learning Finish;
            break;
return w;
```

4. Proposed PPDP scheme

In this section, we will describe the proposed PPDP scheme, which mainly consists three phases: system setting, disease learning and disease prediction.

4.1. PPDP overview

In this part, we first give an overview of the proposed cloud-based privacy-preserving disease prediction system. In reality, suppose a patient is willing to submit his/her symptom information (e.g., blood pressure, heart rate, etc.) to a hospital for disease prediction. Then, the hospital predicts the patient's diseases based on the symptom information and prediction models. In this procedure, the proposed PPDP focuses on how to efficiently train prediction models and use these models to diagnose the diseases without leaking privacy information. Specifically, PPDP mainly consists of two phases: disease learning phase and disease prediction phase.

- **Phase 1: Disease learning phase.** In this phase, the hospital firstly generates ciphertexts of all medical data and prediction models for all diseases, and then submits them

Table 1
Summary of notations.

Algorithm	Symbol	Definition
SLP	x_i	Input vector, $i \in \{1, a\}$, $x_i \in \mathbb{R}^d$
	o_i	Output value, $o_i \in \{-1, 1\}$
	w	Weight vector, $w \in \mathbb{R}^d$
	s	$sign(\cdot)$ function
	η	Learning rate, $\eta > 0$
PPDP	m	Number of diseases, $m > 0$
	D_k	k-disease, $k \in \{1, m\}$
	x_i	i-EHR, $i \in \{1, n\}$, $x_i \in \mathbb{R}^d$
	x_{ij}	j-symptom of i-EHR, $j \in \{1, d\}$, $x_{ij} \in \mathbb{R}$
	w_k	k-disease's prediction model, $w_k \in \mathbb{R}^d$
	o_i	Output of i-EHR, $o_i \in \{-1, 1\}$
	M_1, M_2, H	Secret keys, $\{M_1, M_2\} \in \mathbb{R}^{d\times d}$, $H \in \mathbb{R}^d$
	E_i, F_k	d-dimension random vectors, $\{E_i, F_k\} \in (\mathbb{R}^+)^d$
	C_i, C_{hi}	Ciphertexts of i-EHR, $\{C_i, C_{hi}\} \in \mathbb{R}^{d\times d}$
	C_{wk}	Ciphertext of k-disease's prediction model, $C_{wk} \in \mathbb{R}^{d\times d}$

to the cloud server. After receiving ciphertexts from the hospital, the cloud server executes training procedure based on the encrypted medical information and decides whether to update the medical data. When the hospital receives the updated medical data, it will decrypt and update the prediction models, and then return the prediction model to the cloud in ciphertext. The above training phase starts with a random initialization of the prediction model for each disease, and is then iteratively conducted. At last, the cloud server can obtain encrypted prediction models for all diseases.

- **Phase 2: Disease prediction phase.** When a patient wants to identify his/her disease, a new EHR (i.e., an EHR without the diagnosis result) is submitted to the hospital. Then, the hospital encrypts it and sends the ciphertext to the cloud. Based on the encrypted prediction models, the cloud is able to calculate the prediction results without decrypting the ciphertexts, and then sends the results to the hospital. When the hospital receives prediction results, it returns the results and some medical advices (e.g., disease prediction, outpatients registration, doctor recommendations, etc.) to the patient.

4.2. System setting

For a cloud-based DPS, the hospital bootstraps the whole PPDP scheme. Specifically, the hospital stores a collection of specific disease patterns $\langle D_k\rangle_{k=1}^m$, each of which contains a set of EHRs $\langle x_i, o_i\rangle_{i=1}^n$. x_i is a d-dimension vector, where each element in x_i represents a specific symptom (e.g., temperature, blood pressure, serum cholesterol, etc.) and $o_i \in \{-1, 1\}$ represents the desired output where -1 represents suffering from D_k and 1 represents not. We assume the EHRs have been pre-processed such that the representations are fit for our scheme.

Given two random $d \times d$ invertible matrices $\{M_1, M_2\}$ and a random d-dimension vector H as secret keys, where $\{M_1, M_2\} \in \mathbb{R}^{d\times d}$ and $H \in \mathbb{R}^d$. The hospital encrypts EHRs and the weight as follows.

EHR Encryption: For an EHR $x_i \in D_k$, the hospital generates a random matrix D_i to blind x_i as

$$D_i = \begin{bmatrix} A_{11} * x_{i1} & A_{12} * x_{i1} & \cdots & A_{1d} * x_{i1} \\ A_{21} * x_{i2} & A_{22} * x_{i2} & \cdots & A_{2d} * x_{i2} \\ \vdots & \vdots & \ddots & \vdots \\ A_{d1} * x_{id} & A_{d2} * x_{id} & \cdots & A_{dd} * x_{id} \end{bmatrix} \quad (4)$$

where A_l is set as a d-dimension random vector, $1 \le l \le d, A_l \in \mathbb{R}^d$, and satisfies $\sum_{j=1}^d A_{lj} * H_j = 1$. In other words, x_i can be recovered by computing $x_i^T = D_i \times H^T$.

Table 2
EHRs and ciphertexts for a disease D_k.

EHR	Information	Ciphertexts	Desired output
x_1	$[x_{11}, x_{12}, \ldots, x_{1d}]$	$\{C_1, C_{h1}\}$	o_1
x_2	$[x_{21}, x_{22}, \ldots, x_{2d}]$	$\{C_2, C_{h2}\}$	o_2
$\cdots$	$\cdots$	$\cdots$	$\cdots$
x_n	$[x_{n1}, x_{n2}, \ldots, x_{nd}]$	$\{C_n, C_{hn}\}$	o_n

Then, the hospital encrypts D_i and H as

$$C_i = M_1^{-1} \times D_i \times M_2, \quad (5)$$

$$C_{hi} = M_2^{-1} \times H^T \times E_i \quad (6)$$

where E_i is a random vector with d positive elements, $E_i \in (\mathbb{R}^+)^d$.

The encrypted EHRs are stored as shown in Table 2.

Afterward, the tuple $\{C_i, C_{hi}, o_i, I_k\}$ is uploaded to the cloud server, where I_k is the index of disease D_k.

Weight Encryption: In our scheme, the prediction model (i.e., weight) is also sensitive. For a specific disease D_k, the hospital randomly generates a weight $w_k = [w_{k1}, w_{k2}, \ldots, w_{kd}]$ and encrypts it as

$$C_{wk} = F_k^T \times w_k \times M_1 \quad (7)$$

where F_k is a random vector with d positive elements like E_i, $F_k \in (\mathbb{R}^+)^d$.

Note that, each matrix multiplication has a time complexity of $O(d^3)$ and each matrix–vector multiplication costs $O(d^2)$, where d is the dimension of the EHR vector. Thus, encrypting x_i and w_k costs $O(d^3)$ and $O(d^2)$ respectively.

4.3. Disease learning

The privacy-preserving SLP learning procedure is described by Algorithm 2.

In system setting phase, for each historical EHR $\langle x_i, o_i\rangle_{i=1}^n \in D_k$, where $1 \le k \le m$, the hospital encrypts it and stores the ciphertext $\{C_i, C_{hi}, o_i, I_k\}$ in the cloud server. To learn the prediction model w_k for D_k, the cloud server selects ciphertexts with the same I_k and executes the training process as follows.

- Step 1: The hospital generates a random weight w_k in which not all elements are equal to 0, and executes the *Weight Encryption* operation. Then, the hospital uploads the tuple $\{C_{wk}, I_k\}$ to the cloud server.
- Step 2: With the weight tuple, the cloud server randomly selects a ciphertext $\{C_i, C_{hi}, o_i, I_k\}$ and executes the $sign(\cdot)$ function as

$$s_i = sign(tr(C_{wk} \times C_i \times C_{hi})) \quad (8)$$

where $tr(\cdot)$ denotes the *trace* function. If $s_i \neq o_i$, the cloud server will update the ciphertext C_i as ηC_i and sends it back to the hospital. Thus, in step 2, 2 matrix multiplication operations are needed for the cloud server, each of which costs $O(d^3)$ respectively.

- Step 3: After receiving ηC_i from the cloud server, the hospital updates w_k as

$$w_k = w_k + (M_1 \times \eta C_i \times M_2^{-1} \times H^T)^T \times o_i. \tag{9}$$

Then, the hospital encrypts the improved weight and repeats the steps from step 2. Otherwise, if s_i is equal to o_i, the cloud server will remain w_k and repeat the steps from step 2. In step 3, to update the weight vector, the hospital needs to execute vector–matrix multiplication operations which cost $O(d^2)$.

- Step 4: If the convergence criterion is satisfied, the hospital terminates the training process. After iterating and updating, w_k is identified and can be seen as the prediction model for D_k. The hospital then encrypts w_k as $\{C_{wk}, I_k\}$ and stores it in the cloud server.
- Step 5: After obtaining the prediction model for D_k, the cloud server selects another EHRs training set $\langle x_i, o_i\rangle_{i=1}^n \in D_{k+1}$ and repeats the steps from step 1. After all EHRs are trained, the cloud server gets a set of encrypted prediction models $\langle C_{wk}, I_k\rangle_{k=1}^m$ for disease patterns $\langle D_k\rangle_{k=1}^m$.

Algorithm 2: Privacy-Preserving SLP Learning Algorithm.

Input: n input sample d-dimension EHRs, $< x_i >_{i=1}^n \in D_k$, $1 \leq k \leq m$, $iteration_{max}$, learning rate η, desired value $o_i \in \{-1, 1\}$, sign function $s_i = sign(f(w^T x_i))$;

. **Output**: Prediction model: w_k, $1 \leq k \leq m$;

for $1 \leq k \leq m$ **do**
  the hospital selects historical EHRs set D_k;
  for $1 \leq i \leq n$ **do**
    //the hospital executes *EHR Encryption*;
    x_i is encrypted as $\{C_i, C_{hi}, o_i, I_k\}$ and then uploaded to the cloud server (see Eqs. (4)–(6));
for $1 \leq k \leq m$ **do**
  the hospital randomly initializes w_k ;
  for $iteration = 1, 2 \cdots, iteration_{max}$ **do**
    for $1 \leq i \leq n$ **do**
      //the hospital executes *Weight Encryption*;
      w_k is encrypted as $\{C_{wk}, I_k\}$ and then uploaded to the cloud server (see Eq. (7));
      the cloud server computes $sign(\cdot)$ function (see Eq. (8));
      if $s_i \neq o_i$ **then**
        he cloud server returns ηC_i to the hospital and then the hospital updates the weight (see Eq. (9));
      else
        //Learning Finish;
        break;
return w_k, $1 \leq k \leq m$;

Correctness Analysis: The correctness of disease learning algorithm can be illustrated as follows: In step 2, the cloud server executes the $sign(\cdot)$ function as

$$\begin{aligned} s_i &= sign(tr(C_{wk} \times C_i \times C_{hi})) \\ &= sign\left(tr\left(F_k^T \times w_k \times M_1 \times M_1^{-1} \times D_i \right.\right. \\ &\quad \left.\left. \times M_2 \times M_2^{-1} \times H^T \times E_i\right)\right) \\ &= sign(tr(F_k^T \times w_k \times x_i^T \times E_i)) \\ &= sign(tr(F_k^T \times (w_k \times x_i^T) \times E_i)) \end{aligned} \tag{10}$$

where $w_k \times x_i^T$ is a real number. Note that F_k and E_i are two random positive vectors. Assuming $F_k = [F_{k1}, F_{k2}, \ldots, F_{kd}]$ and $E_i = [E_{i1}, E_{i2}, \ldots, E_{id}]$, by the definition of *trace*, we have

$$T_i = tr(F_k^T \times E_i) = \sum_{j=1}^{d} F_{kj} * E_{ij}. \tag{11}$$

Based on Eq. (11), T_i is always positive and will not compromise the result of $sign(w_k \times x_i^T)$. Thus, the hospital has

$$\begin{aligned} s_i &= sign(tr(C_{wk} \times C_i \times C_{hi})) \\ &= \boldsymbol{sign(w_k \times x_i^T)}. \end{aligned} \tag{12}$$

Thus, the computation result s_i in Eq. (10) is consistent with that in Eq. (1).

In step 3, the hospital updates w_k as

$$\begin{aligned} w_k &= w_k + (M_1 \times \eta C_i \times M_2^{-1} \times H^T)^T \times o_i \\ &= w_k + (\eta D_i \times H^T)^T \times o_i \\ &= \boldsymbol{w_k + \eta x_i o_i} \end{aligned} \tag{13}$$

which is also consistent with that in Eqs. (2) and (3).

Thus, the hospital can get the correct prediction model in the outsourcing environment. In other words, the correctness of disease learning is satisfied.

4.4. Disease prediction

In disease prediction phase, for m prediction models $\langle w_k\rangle_{k=1}^m$, the cloud server has already encrypted and stored the ciphertexts in the cloud. When a patient submits a new EHR, the prediction operations are executed as follows.

- Step 1: With the new coming EHR x_q, the hospital encrypts it as $\{C_q, C_{hq}\}$ following *EHR Encryption* as

$$C_q = M_1^{-1} \times D_q \times M_2, \tag{14}$$

$$C_{hq} = M_2^{-1} \times H^T \times E_q. \tag{15}$$

Then, the hospital uploads the ciphertext to the cloud server. In this step, the time cost of hospital is the same as that of encrypting x_i in the system initialization phase, which is $O(d^3)$.

- Step 2: On receiving $\{C_q, C_{hq}\}$, the cloud server executes the $sign(\cdot)$ function between the encrypted prediction model $\langle C_{wk}, I_k\rangle_{k=1}^m$ and $\{C_q, C_{hq}\}$ as

$$s_k = sign(tr(C_{wk} \times C_q \times C_{hq})). \tag{16}$$

If $s_k > 0$, the cloud server considers the patient may suffer from the disease and sends the corresponding I_k to the hospital. The cloud server needs to execute 2 matrix multiplication operations, each of which costs $O(d^3)$ respectively.

- Step 3: After receiving the index, the hospital returns the prediction results to the patient.

Correctness Analysis: The correctness of disease prediction can be illustrated as follows: In step 2, the cloud server executes $sign(\cdot)$ as

$$\begin{aligned} s_k &= sign(tr(C_{wk} \times C_q \times C_{hq})) \\ &= sign(tr(F_k^T \times E_q \times w_k \times x_q^T)) \\ &= \boldsymbol{sign(w_k \times x_q^T)}. \end{aligned} \tag{17}$$

Based on the above equation and the definition of SLP, the cloud server can confirm if patients suffer from the disease of D_k by computing s_k. As a result, the correctness of disease prediction is satisfied.

5. Security analysis

In this section, we analyze the security of the proposed PPDP scheme. As mentioned in Section 2.2, a higher-level attack is more powerful than a lower-level one. Thus, if the PPDP scheme can resist *Level-III* attack, we can demonstrate that it is secure. Specifically, in *Level-III* attack, the adversary can (i) observe the ciphertexts $\{C_i, C_{hi}, C_{wk}\}$ and all computation results, and (ii) observe some historical EHRs in the plaintexts database and know the corresponding ciphertexts, and (iii) choose some queries of interest and observe their encrypted versions. Our analysis focuses on how the proposed PPDP achieves privacy protection of EHRs and prediction model (i.e., weight) under *Level-III* attack.

Theorem 1. *In the proposed PPDP scheme, EHRs are privacy-preserving under Level-III attack.*

Proof. Let the knowledge of the adversary be a set of tuples $\langle x_i, C_i, C_{hi}\rangle$. Without loss of generality, the adversary can choose any t plaintext/ciphertext pairs in the EHRs database and observe C_{wk} in the training phase.

As shown in Eqs. (5)–(7), to recover x_i, the adversary needs to recover the secret keys M_1, M_2 and H. Accordingly, the attacker can build the following equations as

$$\begin{aligned} C_i &= M_1^{-1} \times D_i \times M_2, \\ C_{hi} &= M_2^{-1} \times H^T \times E_i, \\ C_{wk} &= F_k^T \times w_k \times M_1. \end{aligned} \tag{18}$$

In these equations, $M_1, M_2, M_1^{-1}, M_2^{-1}$ are $d*d$ random unknown matrices and H^T is a random d-dimension vector. Note that D_i is a random $d*d$ matrix which is blinded by d^2 random unknowns, and E_i, F_k^T are chosen differently in each encryption operation. Thus, all ciphertexts in the proposed PPDP are randomly generated.

To recover M_1, $2d^2$ equations can be built. However, with $(2d^2 + 2d)$ unknowns, the adversary cannot compute M_1. Following the same analysis, the adversary cannot recover M_2 with $2d^2$ equations and $(2d^2 + 2d)$ unknowns. To recover H^T, d equations can be built, but there are $(d^2 + d)$ unknowns. Therefore, the adversary cannot recover secret keys M_1, M_2 and H with the ciphertexts and the corresponding plaintexts.

While the secret keys cannot be recovered, the adversary may propose to bypass the computation of secrets and derive x_i directly. In disease learning phase, the adversary can compute $C_{wk} \times C_i \times C_{hi}$, denoted as P_i,

$$\begin{aligned} P_i &= C_{wk} \times C_i \times C_{hi} \\ &= F_k^T \times w_k \times x_i^T \times E_i \end{aligned} \tag{19}$$

where F_k^T and E_i are random positive vectors. To recover x_i^T, the adversary can build d equations, however, x_i^T cannot be computed, since $3d$ unknowns are included in these equations. When the adversary acts as a patient, he/she may submit multiple prediction queries, say u queries. Thus, the adversary is able to build $u * d$ more equations according to Eq. (19). However, $u * 3d$ unknowns are also introduced along with the u queries. Moreover, we assume that an "ideal" powerful attacker knows w_k (which is impossible in practice). However, although d more equations can be built, the number of equations are still fewer than the number of unknowns.

Therefore, based on the above analysis, EHRs are well protected and the *Level-III* attacker cannot build enough knowledge to break the PPDP scheme.

Theorem 2. *In the proposed PPDP scheme, prediction model achieves privacy-preserving under Level-III attack.*

Proof. The prediction model is a d-dimension vector randomly generated by the hospital and updated in the training process. Following the same analysis in proof, secret keys are well protected. Then, we consider $\langle w_k, C_{wk}\rangle$. As shown in Eqs. (7) and (19), to recover w_k, the adversary can build the following equations

$$C_{wk} = F_k^T \times w_k \times M_1 \tag{20}$$

$$P_i = F_k^T \times w_k \times x_i^T \times E_i \tag{21}$$

where F_k and E_i are randomly generated by the hospital. To recover w_k, the adversary can build $2d$ equations. However, w_k cannot be computed with $2d$ equations and $(d^2 + 3d)$ unknowns.

Then, we assume x_i and x_q are known to the adversary (e.g., the adversary observes x_i in the plaintexts database or acts as a patient to submit x_q). Although the adversary holds x_i as his/her extra knowledge, there are $(d^2 + 2d)$ unknowns in $2d$ equations.

We further consider an "ideal" attacker who has the ability to observe some weights. To recover the unknown w_k, the attacker needs to figure out the secret keys. However, note that F_k is randomly generated when encrypting w_k. With d equations built, another d unknowns will be introduced. Therefore, the *Level-III* attacker cannot access the sensitive information.

6. Performance evaluation

In this section, we present the performance evaluation of the proposed PPDP scheme.

6.1. Computation cost

We evaluate computation cost of the proposed PPDP scheme by using Java language. Specifically, for hospital, we utilize a laptop with Core(TM) i5-2430M 2.40 GHz CPU and 8GB memory, and use the same laptop for the cloud server. In the experiment, two datasets are considered. A real dataset [25] is utilized from the UCI machine learning repository. This dataset is used to test the performance of the SLP classifier based on our PPDP scheme. Without loss of generality, we also utilize a synthetic dataset to test all the factors which might affect the performance of PPDP.

6.1.1. Real dataset

The real dataset includes 683 instances for Breast Cancer and 297 instances for Heart Disease. For Breast Cancer dataset, each instance contains nine attributes [clump thickness; uniformity of cell size; uniformity of cell shape; marginal adhesion; single epithelial cell size; bare nuclei; bland chromatin; normal nucleoli; mitoses] and two decisions [benign as 1; malignant as -1]. All these attributes are integers from 1 to 10. For Heart Disease dataset, each instance contains thirteen attributes [age; sex; chest pain type; trestbps; chol; fbs; restecg; thalach; exang; oldpeak; slope; ca; thal] and also two decisions [benign as 1; malignant as -1]. All attributes range from 0 to 10 except for age, trestbps (resting blood pressure), chol (serum cholesterol in mg/dl) and thalach (maximum heart rate achieved). We first use the proposed PPDP to train SLP classifier, and then use the SLP classifier and two real datasets to test the error rate of the classifier. We set the iteration threshold as 10 000 and the specific information of two real datasets is listed in Table 3. From this table, we can see the error rate for Breast Cancer and Heart Disease are around 16.83% and 16.13% respectively, which is acceptable [6]. We also test the running efficiency about PPDP. Table 4 shows the time costs of PPDP, HE-Based and Local schemes in different phases after 10 000 iterations. For Breast Cancer dataset, it takes 0.011 s to

Table 3
Experiment datasets and parameters.

Dataset	Sample	Attributes	Class	Learning rate	Error rate
Breast cancer	683	9	2	0.1	16.83%
Heart disease	297	13	2	0.1	16.13%

Table 4
Time costs of PPDP, HE-based and local schemes.

Dataset	Phase	PPDP	HE-based scheme	Local scheme
Breast cancer	Data encryption	0.011 s	54.308 s	0 s
	Disease learning	0.058 s	3069.824 s	0.015 s
Heart disease	Data encryption	0.013 s	85.155 s	0 s
	Disease learning	0.142 s	1764.279 s	0.013 s

encrypt all instances, and in disease learning phase, it needs 0.058 s (0.025 s spent on hospital side and 0.031 s spent on the cloud server side). For Heart Disease dataset, it takes 0.013 s to encrypt all 297 instances, and in disease learning phase, it requires 0.140 s (including 0.071 s for hospital and 0.068 s for the cloud server).

To show the efficiency of PPDP, we implement a privacy-preserving SLP algorithm based on homomorphic encryption technique, called HE-Based scheme, where the modulus is set as 1024 bits and at least $1 - 2^{-64}$ certainty of primes are generated, and a scheme in which the hospital performs the SLP algorithm by itself without privacy protection, called local scheme. From Table 4, we can see PPDP takes less than one second to execute data encryption and disease learning, while the HE-Based scheme needs tens of seconds and thousands of seconds respectively. Therefore, the running time of PPDP is significantly less than that of HE-Based scheme. In disease learning phase, local scheme needs 0.0015 s and 0.0013 s for Breast Cancer and Heart Disease respectively. In contrast, PPDP needs 0.058 s and 0.142 s. The reason is though the cloud server can efficiently perform computation over ciphertexts, the computational overhead of data encryption/decryption on the hospital side is inevitable. However, we emphasize that the local scheme has sacrificed substantial security and thus cannot be utilized to train medical data which are stored in the cloud server.

6.1.2. Synthetic database

We randomly generate synthetic dataset, which consists of 500 tuples with 20 attributes. All attributes are randomly picked from 0 to 10. The number of iteration is set as 10 000. Note that there are three factors that affect the total running time of PPDP: the number of historical EHRs (N_{hehr}), the number of symptom attributes contained in each medical data (N_{sa}), the number of diseases (N_d). In Figs. 3–5, we plot the running time over the synthetic dataset of the PPDP and HE-Based schemes vary with N_{hehr}, N_{sa} and N_d respectively.

In Fig. 3(a)-(b), the data encryption time increases with the number of medical data, where $50 \leq N_{hehr} \leq 500$, $N_d = 1$, $N_{sa} = 20$. The reason is the hospital needs to encrypt more data as N_{hehr} increases. As can be seen, when N_{hehr} reaches 500, PPDP only needs 48 ms to encrypt medical data, while the HE-Based scheme takes 139.6 s. In disease learning phase, the running time is nearly not affected, since the number of iteration is stable. For the two schemes, it takes about 500 ms and 7000 s respectively, which demonstrates the efficiency of PPDP.

In Fig. 4(a)–(b), we show the running time varies with the number of symptom attributes, where $11 \leq N_{sa} \leq 20$, $N_d = 1$, $N_{hehr} = 500$. As can be seen, the running time of data encryption and disease learning increases with the number of attributes. With the increase of N_{sa}, PPDP needs to perform higher-dimension matrices operations, and the HE-Based scheme calls more multiplications. For PPDP, the data encryption time ranges from 13 ms to 50 ms as the number of attributes varies from 11 to 20. For HE-Based scheme, the time cost ranges from 77.787 s to 141.062 s. Thus, PPDP can significantly improve the efficiency in data encryption and disease learning phase.

Table 5
A summary of computation cost for each EHR x_i in PPDP, $d \ll s$, $d \ll m$.

Phase	Step	Entity	Computation cost
System setting	–	Hospital	$O(s * d^3)$
Disease learning	Step 2	Cloud	$O(d^3)$
		Hospital	$O(d^2)$
Disease prediction	Step 1	Hospital	$O(d^3)$
	Step 2	Cloud	$O(m * d^3)$

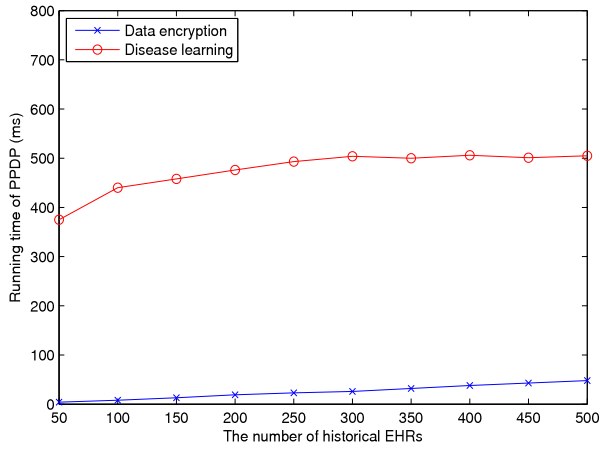
In Fig. 5(a)–(b), we plot the running time in disease prediction phase varies with the number of diseases, where $50 \leq N_d \leq 200$, $N_{sa} = 20$. It can be seen, for a new coming EHR, the running time at the cloud side increases linearly with increased number of diseases, while the running time at the hospital side remains approximately constant. The reason is, in disease prediction phase, the hospital only needs to encrypt the new coming EHR, while the cloud calls more operations (i.e., $sign(tr(C_{wk} \times C_q \times C_{hq}))$ for PPDP, and $E(w) \cdot E(x_i)$ for HE-Based scheme). When the number of diseases varies from 50 to 500, the prediction time ranges from 0.52 ms to 4.298 ms for PPDP, and ranges from 2.170 s to 17.524 s for the HE-Based scheme. That is, PPDP can save about 99.98% time for disease prediction.

6.2. Complexity analysis

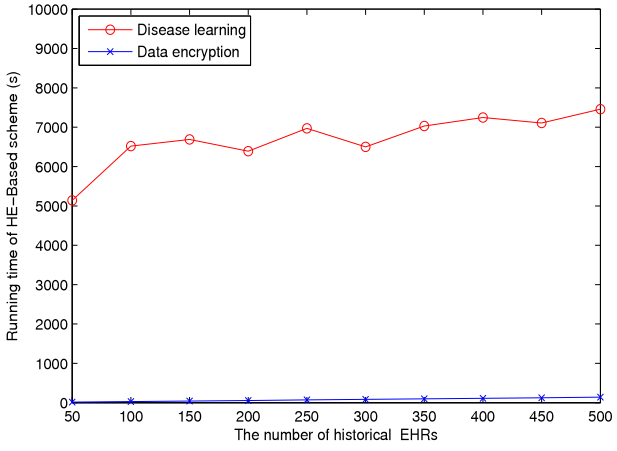
In this part, we give an overview of the complexity of the proposed PPDP scheme, in terms of the computation and communication costs.

6.2.1. Computation cost

Table 5 illustrates the computation cost for each step of system setting, disease learning and disease prediction phases. Note that, matrix multiplication has a time complexity of $O(d^3)$, where d is the dimension of the EHR vector, and matrix–vector multiplication costs $O(d^2)$. In the system setting phase, encrypting x_i and H costs $O(d^3)$ and $O(d^2)$ respectively. Thus, the complexity for the system setting is $O(s * d^3)$, where s is the number of EHRs in the database. Although the complexity grows linearly with the number of EHRs, it is a one-time cost. In step 2 of training phase, to execute $sign(\cdot)$ function, 2 matrix multiplications are needed for the cloud server, each of which costs $O(d^3)$. On the hospital side, updating the weight costs $O(d^2)$. In the disease prediction phase, the time cost of step 1 is the same as that of encrypting x_i in the system initialization. To diagnose patients' diseases, in step 2, the complexity for the cloud server is $O(m * d^3)$, where m is the number of prediction models.

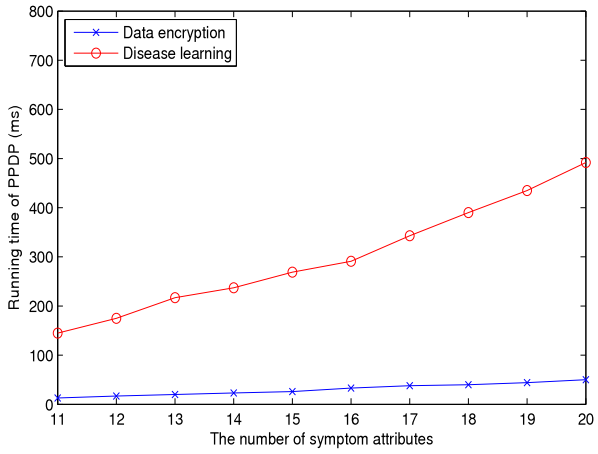


(a) PPDP scheme.

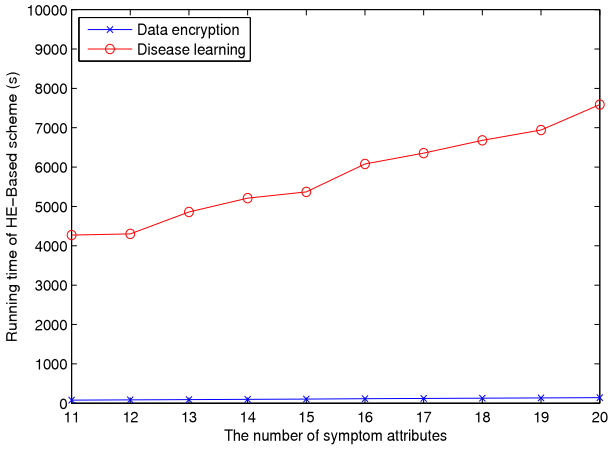


(b) HE-based scheme.

Fig. 3. Computation cost varying with the number of historical EHRs.



(a) PPDP scheme.



(b) HE-based scheme.

Fig. 4. Computation cost varying with the number of symptom attributes.

Table 6
A summary of communication overhead in PPDP. Here, k denotes the number of disease prediction results returned, $k \ll m$.

Message	Communication overhead
$E(x_i)$	$O(s * d^2)$
C_{wk}	$O(d^2)$
ηC_i	$O(d^2)$
I	$O(k)$

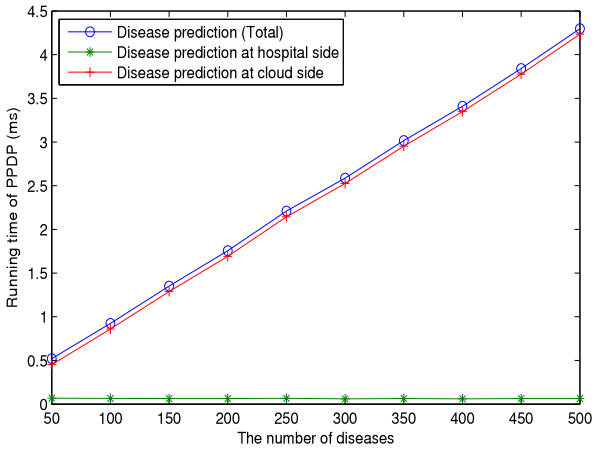
6.2.2. Communication cost

Table 6 illustrates the communication overhead of the proposed PPDP scheme. As described in Section 4, the transmitted data includes the encrypted EHRs $E(x_i)$, encrypted weight C_{wk}, updated data ηC_i and the prediction results I. In system setting phase, it costs $O(s * d^2)$ to transmit the whole encrypted database from the hospital to the cloud server, where s is the number of EHRs. Although the communication time grows linearly with the size of EHRs database, it is a one-time cost for system setting. To reduce the bandwidth cost, hard disks can be used to transmit the encrypted database. For other transmitted data, C_{wk} is a $d \times d$ matrix which costs $O(d^2)$, ηC_i is a $d * d$ matrix with a cost of $O(d^2)$, and prediction results are indexes which costs $O(k)$.

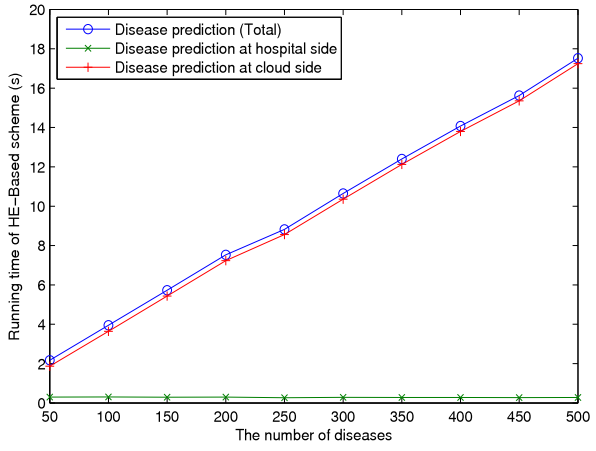
7. Related work

Recently, a large variety of disease prediction models have been developed in biomedical engineering [7–13]. For example, to diagnose the neurological diseases, multiclass support vector machine was utilized for multiclass electroencephalogram signals [7]. Ajemba et al. [8] proposed a fast predictive model by using a support vector classifier approach to predict the risk of progression of adolescent idiopathic scoliosis. In order to diagnose the pancreatic cancer, Wang et al. [9] developed a disease prediction approach by using Bayesian classification. In [10], Barakat et al. proposed a hybrid system by using SVM for the diagnosis of diabetes. A diagnosis model was constructed by Huang et al. for the diagnosis of breast cancer by using SVM [11]. To predict the heart disease, Anooj et al. [12] proposed a fuzzy rule-based decision support system. Focus on multivariate logistic regression, Bouwmeester et al. [13] developed a prediction model using multiple symptoms and environmental data to fit a logarithmic transformation of the likelihood of the tested disease. These works, though have developed various prediction models, fail to take into consideration an important issue in the design of disease prediction systems, i.e., *the privacy protection of medical information*, especially privacy has become a major concern in nowadays.

To address this challenge, *secure disease prediction* [3–6,14–16], i.e., to diagnose patients' diseases without leaking medical data and prediction model, have recently been widely studied. Mathew and Obradovic [14] presented a privacy-preserving framework for building a clinical tool in the form of decision tree, nevertheless, it can only protect the privacy of training database by employing statics about the samples. Bos et al. [4] implemented



(a) PPDP scheme.



(b) HE-based scheme.

Fig. 5. Computation cost for a new EHR prediction varying with the number of diseases.

a privacy-preserving prediction service to diagnose the possibility of suffering from a disease based on the logistic regression and Cox proportional model by using a lattice-based homomorphic encryption scheme [26]. Wang et al. [15] presented a cloud-based privacy-preserving Single-Layer Perceptron learning for e-Healthcare based on the Paillier homomorphic encryption technique. However, for the two schemes in [4] and [15], the prediction model is publicly known. Focus on the protection of human data, Wang et al. [6] proposed a Healer framework. Specifically, they used a small samples size to facilitate secure rare variants analysis, and obtained final results by decrypting ciphertexts in the trusted party. In their framework, somewhat homomorphic encryption technique is utilized to protect sensitive information. Rahulamathavan et al. [5] presented a privacy-preserving scheme by using support vector machine for the diagnosis of patients' diseases without disclosing any privacy of patients. Similarly, Liu et al. [3] proposed a privacy-preserving clinical diagnosis system by using Naïve Bayesian classification, which can help clinicians to securely diagnose the risk of patients' diseases. These works, since all the encryption and decryption operations are based on homomorphic encryption technique, are not efficient. To improve the efficiency, Zhu et al. [16] proposed a novel framework that greatly improves the prediction efficiency without disclosing any sensitive medical information. In their protocol, medical data and support vectors are protected with lightweight multi-party random masking and polynomial aggregation technique based on an improved expression for the nonlinear SVM. However, their work is based on the assumption that there is no collusion between the patient and the server provider since all support vectors are stored in the server in the plaintext form. Once such a collusion occurs, the components of the SVM classifier can be computed and then the diagnosis model will be disclosed. To improve the efficiency of big data feature learning, Zhang et al. [27] proposed a privacy-preserving deep computation model on cloud for big data feature learning, which might be utilized in disease prediction system. However, the accuracy performance of their scheme is a little lower than that of the deep computation model without privacy protection [28].

Recently, encryption algorithms based on matrices has received increasing attention and has been utilized in many fields. Accordingly, a series of works such as [29–34] have been published. Different from privacy-preserving disease prediction schemes based on traditional encryption techniques such as Paillier homomorphic encryption techniques, this paper utilizes the matrices to encrypt data and predict patients' diseases without leaking the privacy information. Compared with existing works based on some sophisticated machine learning algorithms such as SVM [16], Naïve Bayesian classification [3], this paper uses SLP to build prediction models. SLP is more efficient due to its simplicity. Besides, the prediction result is acceptable compared with the schemes in [3,15,16].

8. Conclusions

In this work, we propose an efficient and privacy-preserving disease prediction scheme, called PPDP. The proposed PPDP scheme is characterized by employing random vectors and matrices, which enables the outsourced EHRs can be handled and trained on the cloud server by using SLP algorithm without leaking sensitive information. Detailed analysis indicates that the proposed PPDP achieves a high level of privacy protection and efficiency. In future, we will work on designing more efficient and privacy-preserving disease prediction models.

Acknowledgment

This research is supported by the National Natural Science Foundation of China (Grant Nos. 61402037, 61272512).

References

[1] Healthcare needs moonshots. This is what a digital one looks like, 2016. [Online]. Available: http://newsroom.gehealthcare.com/healthcare-needs-moonshots-this-is-what-a-digital-one-looks-like/.
[2] M. Li, S. Yu, K. Ren, W. Lou, Securing personal health records in cloud computing: patient-centric and fine-grained data access control in multi-owner settings, in: International Conference on Security and Privacy in Communication Systems, 2010, pp. 89–106.
[3] X. Liu, R. Lu, J. Ma, L. Chen, B. Qin, Privacy-preserving patient-centric clinical decision support system on naive Bayesian classification, IEEE J. Biomed. Health Inform. 20 (2) (2016) 655–668.
[4] J.W. Bos, K. Lauter, M. Naehrig, Private predictive analysis on encrypted medical data, J. Biomed. Inform. 50 (2014) 234–243.
[5] Y. Rahulamathavan, S. Veluru, R.C.W. Phan, J. Chambers, M. Raiarajan, Privacy-preserving clinical decision support system using gaussian kernel-based classification, IEEE J. Biomed. Health Inform. 18 (1) (2014) 56–66.
[6] S. Wang, Y. Zhang, W. Dai, K. Lauter, M. Kim, Y. Tang, H. Xiong, X. Jiang, HEALER: Homomorphic computation of ExAct Logistic rEgRession for secure rare disease variants analysis in GWAS, Bioinformatics 32 (2) (2016) 211–218.
[7] I. Fuler, E.D. Uberli, Multiclass support vector machines for EEG-signals classification, IEEE Trans. Inform. Technol. Biomed. 11 (2) (2007) 117–126.
[8] P. Ajemba, L. Ramirez, N. Durdle, D. Hill, V. Raso, A support vectors classifier approach to predicting the risk of progression of adolescent idiopathic scoliosis, IEEE Trans. Inform. Technol. Biomed. 9 (2) (2005) 276–282.
[9] W. Wang, S. Chen, K.A. Brune, R.H. Hruban, G. Parmigiani, A.P. Klein, PancPRO: risk assessment for individuals with a family history of pancreatic cancer, J. Clin. Oncol. 25 (11) (2007) 1417–1422.

[10] M.N.H. Barakat, A.P. Bradley, Intelligible support vector machines for diagnosis of diabetes mellitus, IEEE Trans. Inform. Technol. Biomed. 14 (4) (2010) 1114–1120.
[11] C.L. Huang, H.C. Chen, M.C. Chen, Prediction model building and feature selection with support vector machines in breast cancer diagnosis, Expert Syst. Appl. 34 (1) (2008) 578–587.
[12] P.K. Anooj, Clinical decision support system: Risk level prediction of heart disease using weighted fuzzy rules, J. King Saud Univ.-Comput. Inf. Sci. 24 (1) (2012) 27–40.
[13] W. Bouwmeester, J.W.R. Twisk, T.H. Kappen, et al., Prediction models for clustered data: comparison of a random intercept and standard regression model, BMC Med. Res. Methodol. 13 (1) (2013) 19.
[14] G. Mathew, Z. Obradovic, A privacy-preserving framework for distributed clinical decision support, in: Computational Advances in Bio and Medical Sciences, ICCABS. 2011, pp. 129–134.
[15] G. Wang, R. Lu, C. Huang, PSLP: Privacy-preserving single-layer perceptron learning for e-Healthcare, in: 10th International Conference on Information, Communications and Signal Processing, ICICS. 2015, pp. 1–5.
[16] H. Zhu, X. Liu, R. Lu, H. Li, Efficient and privacy-preserving online medical pre-diagnosis framework using nonlinear SVM, IEEE J. Biomed. Health Inform. (2016).
[17] P. Guo, J. Wang, S. Ji, X. Geng, N. Xiong, A lightweight encryption scheme combined with trust management for privacy-preserving in body sensor networks, J. Med. Syst. 39 (12) (2015) 1–8.
[18] A. Wang, J. Liu, X. Wang, J. Wang, A fuzzy control theory and neural network based sensor network control system, in: International Conference on Intelligent Information Hiding and Multimedia Signal Processing 665, IH-MSP. 2014, pp. 831–834.
[19] Z. Shi, J. Liu, Q. Song, J. Wang, An energy efficient data transmission mechanism for middleware of wireless sensor network, in: Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2014, pp. 827–830.
[20] S. di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, P. Samarati, Over-encryption: management of access control evolution on outsourced data, in: Proceedings of the 33rd international conference on very large data bases, 2007, pp. 123–134.
[21] Q. Wang, S. Hu, K. Ren, CloudBI: Practical privacy-preserving outsourcing of biometric identification in the cloud, in: European Symposium on Research in Computer Security, 2015, pp. 186–205.
[22] H. Delfs, H. Knebl, Introduction to Cryptography Principles and Applications, 2002.
[23] K. Liu, C. Giannella, H. Kargupta, An attacker's view of distance preserving maps for privacy preserving data mining, in: European Conference on Principles of Data Mining and Knowledge Discovery, 2006, pp. 297–308.
[24] Y. Freund, R.E. Schapire, Large margin classification using the perceptron algorithm, Mach. Learn. 37 (3) (1999) 277–296.
[25] M. Lichman, UCI machine learning repository, 2013 [Online]. Available: http://archive.ics.uci.edu/ml.
[26] J.W. Bos, K. Lauter, J. Loftus, M. Naehrig, Improved security for a ring based fully homomorphic encryption scheme, in: Cryptography and Coding, Springer, Berlin, Heidelberg, 2013, pp. 45–64.
[27] Q. Zhang, L. Yang, Z. Chen, Privacy preserving deep computation model on cloud for big data feature learning, IEEE Trans. Comput. 65 (5) (2016) 1351–1362.
[28] V. Kulkarni, K. Wagh, Review on privacy preserving deep computation model on cloud for big data feature learning, Int. J. Sci. Res. 12 (5) (2016) 2319–7064. ISSN (Online).
[29] W.K. Wong, D.W. Cheung, B. Kao, N. Mamoulis, Secure kNN computation on encrypted databases, in: Proceedings of the 2009 ACM SIGMOD International Conference on Management of data, ACM, 2009, pp. 139–152.
[30] B. Wang, S. Yu, W. Lou, Y. Hou, Privacy-preserving multi-keyword fuzzy search over encrypted data in the cloud, INFOCOM, 2014, pp. 2112–2120.
[31] B.K. Samanthula, Y. Elmehdwi, W. Jiang, k-Nearest neighbor classification over semantically secure encrypted relational data, IEEE Trans. Knowl. Data Eng. 27 (5) (2015) 1261–1273.
[32] J. Yuan, S. Yu, Efficient privacy-preserving biometric identification in cloud computing, INFOCOM, 2013, pp. 2652–2660.
[33] S. Pan, S. Yan, W. Zhu, Security analysis on privacy-preserving cloud aided biometric identification schemes, in: Australasian Conference on Information Security and Privacy, 2016, pp. 446–453.
[34] Y. Zhu, Z. Wang, J. Wang, Collusion-resisting secure nearest neighbor query over encrypted data in cloud, revisited, Quality of Service, IWQoS, 2016 IEEE/ACM 24th International Symposium on, 2016, pp. 1–6.

Chuan Zhang received the bachelor's degree in network engineering from Dalian University of Technology, Dalian, China, in 2015. He is s currently a graduate student in School of Computer Science and Technology at Beijing Institute of Technology. His research interests include secure data services in cloud computing, security & privacy in VANET and big data security.

Liehuang Zhu received the Ph.D. degree in computer science from Beijing Institute of Technology, Beijing, China, in 2004. He is currently a professor at School of Computer Science & Technology, Beijing Institute of Technology. His research interests include security protocol analysis and design, group key exchange protocol, wireless sensor network, and cloud computing.

Chang Xu received the Ph.D. degree in computer science from Beihang University in 2013, master degree and bachelor degree in School of Computer Science and Technology in Jilin University in 2008 and 2005, respectively. She is currently an assistant professor in School of Computer Science and Technology at Beijing Institute of Technology. Her research interests include security & privacy in VANET, and big data security.

Rongxing Lu received the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Waterloo, Canada, in 2012. He was a Post-Doctoral Fellow with the University of Waterloo from 2012 to 2013. He was an Assistant Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, from 2013 to 2016. He has been an Assistant Professor with the Faculty of Computer Science, University of New Brunswick, Canada, since 2016. His research interest include applied cryptography, privacy enhancing technologies, and IoT-Big Data security and privacy. He currently serves as the Secretary of the IEEE ComSoc CIS-TC. He is currently a Senior Member of the IEEE Communications Society. He received the most prestigious Governor General's Gold Medal and the 8th IEEE Communications Society (ComSoc) Asia Pacific Outstanding Young Researcher Award, in 2013.