Hindawi
Security and Communication Networks
Volume 2022, Article ID 7751845, 17 pages
https://doi.org/10.1155/2022/7751845

Research Article

Practical Privacy Preserving-Aided Disease Diagnosis with Multiclass SVM in an Outsourced Environment

Ruoli Zhao,[1] **Yong Xie**,[1] **Xingxing Jia**,[2] **Hongyuan Wang**,[3] **and Neeraj Kumar**[4,5]

[1]*Department of Computer Technology and Applications, Qinghai University, Xining, China*
[2]*School of Mathematics and Statistics, Lanzhou University, Lanzhou, China*
[3]*Qinghai Province Yindajihuang Project Construction and Operation Bureau, Xining, China*
[4]*School of Computer Science, University of Petroleum and Energy Studies, Dehradun, Uttarakhand, India*
[5]*Department of Computer Science and Information Engineering, Asia University, Taichung, Taiwan*

Correspondence should be addressed to Yong Xie; mark.y.xie@qq.com

Received 20 April 2022; Accepted 20 September 2022; Published 12 October 2022

Academic Editor: Ch. Aswani Kumar

With the rapid development of cloud computing and machine learning, using outsourced stored data and machine learning model for training and online-aided disease diagnosis has a great application prospect. However, training and diagnosis in an outsourced environment will cause serious challenges to the privacy of data. At present, many scholars have proposed privacy preserving machine learning schemes and made a lot of progress, but there are still great challenges in security and low client load. In this paper, we propose a complete privacy preserving outsourced multiclass SVM training and aided disease diagnosis scheme. We design some efficient basic operation algorithms for encrypted data. Then, we design an efficient and privacy preserving SVM model training protocol using the basic operation algorithms. We propose a secure maximum finding algorithm and secure comparison algorithm. Then, we design an efficient online-aided disease diagnosis scheme based on the BFV cryptosystem and blinding technique. Detailed security analysis proves that our scheme can protect the privacy of each participant. The experimental results illustrate that our proposed scheme significantly reduces the computation overhead compared with the previous similar works. Our proposed scheme completes most of the operations of aided disease diagnosis by the cloud servers and the client only needs to complete a small amount of encryption and decryption operations. The overall computation overhead is 0.175 s, and the efficiency of online aided disease diagnosis is improved by 85.4%. At the same time, our proposed scheme provides multiclass diagnosis results, which can better assist doctors in their treatment.

1. Introduction

Machine learning (ML) uses the computer system to build mathematical models on sample data with statistical methods and makes predictions or decisions without being explicitly programmed. Now, ML has shown significant advantages in the field of disease diagnosis and brings more and more convenience to the prevention and treatment of diseases.

With the rapid development of cloud computing technology, cloud service providers (CSP) have high-quality computation and huge storage space, which can provide data processing, model training, diagnosis services and deployment, and other intelligent solutions based on machine learning. In this context, the local clients will outsource their medical data and machine learning models to CSP without having to build their own large-scale infrastructure and computing resources. The cloud can train a machine learning model and provide aided disease diagnosis service by using the outsourced medical data and machine learning models, which can help improve doctors' diagnosis, treatment decisions and provide patients an online disease diagnosis service. A typical cloud platform machine learning system architecture is shown in Figure 1.

However, the security and privacy of outsourced data will be threatened by various threats, making people afraid to

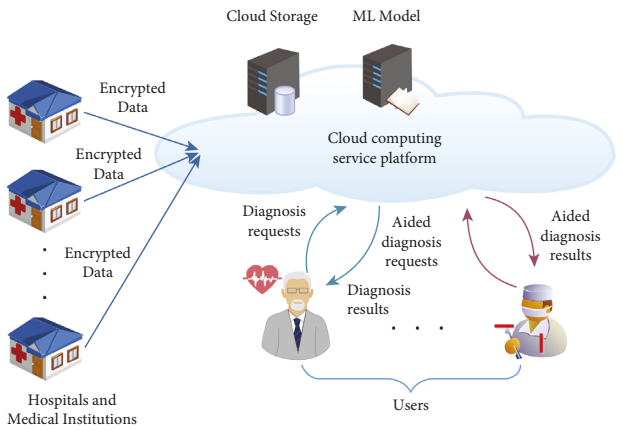

Figure 1: A typical cloud platform machine learning system architecture.

use the service of CSP. The security and privacy threats are mainly reflected in the leakage of the data, the machine learning model of the model owners, the users' request, and diagnosis results. As we all know, the leakage of medical information may cause irreversible losses or become a major event. Therefore, the security and privacy preserving of model training and diagnosis based on cloud computing have become a major challenge.

To address the abovementioned challenges, many scholars have proposed various schemes, such as a secure outsourced classification based on logistic regression model [1], an electronic medical disease risk prediction scheme based on naive Bayes model [2], and other secure disease prediction schemes based on machine learning technology [3–5]. As a machine learning algorithm with high computational efficiency and nice predictive accuracy, the support vector machine (SVM) has achieved high classification accuracy and efficiency in the medical field [6, 7]. However, the existing privacy preserving SVM schemes mainly implement secure prediction [8–11], and there are few privacy preserving SVM schemes for secure training. Most of the existing privacy preserving SVM schemes are designed for binary classification, which can only determine whether the patient has the disease [12], but cannot deal with the multiclass of the disease. In addition, multiclass SVM requires more computation, which will reduce the efficiency [13].

To solve the abovementioned problems, we propose an efficient and privacy preserving online disease diagnosis scheme based on the SVM algorithm. In our scheme, we can achieve multi-class SVM training on the encrypted outsourced data from multiple data owners and provide users with privacy preserving disease diagnosis. In summary, our contributions are as follows:

(1) Efficient and secure basic operation algorithms: Based on the Paillier cryptosystem, we design several basic operation algorithms to realize the secure outsourced data storage and computation, including secure aggregation algorithm, secure multiplication algorithm, and so on. These secure computation algorithms are the building blocks for our proposed training protocol.

(2) Completing machine learning process under privacy preserving: Aiming at the general machine learning process and the goal of privacy preserving, we propose a privacy preserving outsourced multiclass SVM model training and online-aided disease diagnosis scheme. Different from the existing privacy preserving schemes that only support training or diagnosis, our proposed scheme extends the function of privacy preserving machine learning system.

(3) Efficient and secure online aided disease diagnosis: Based on the BFV cryptosystem, we design a secure maximum finding algorithm and secure comparison algorithm. We provide an efficient and privacy preserving aided disease diagnosis scheme. Experimental results illustrate that our proposed scheme significantly reduces the computation cost than the existing similar schemes, which is suitable for practical application scenarios where a large number of users request diagnosis at the same time.

(4) Low overhead for local client: For a local client, the client only needs to perform encryption and decryption operations in our proposed scheme, which reduces the storage and computation overhead of the local client to the greatest extent and makes full use of the computation power of the cloud servers.

The remainder of this paper is organized as follows. In Section 2, we review some related works. In Section 3, we review the Paillier cryptosystem, BFV cryptosystem, and SVM algorithm as preliminaries. In Section 4, we make a system overview. Then, we propose our scheme in Section 5. In Section 6, we analyze the security of our proposed scheme. In Section 7, we make a performance evaluation. Finally, we conclude this paper in Section 8.

2. Related Work

In this section, we summarize the privacy preserving machine learning schemes in recent years.

With the development of big data era, machine learning has been widely used in many fields. Among them, the application of machine learning in the field of intelligent disease diagnosis has developed rapidly. Disease diagnosis schemes based on various machine learning classification algorithms have been proposed [14–17]. However, at the same time, the problem of privacy disclosure in the machine learning process is becoming more and more serious. So, many scholars have carried out the research studies on privacy preserving machine learning.

Triastcyn and Faltings [18] proposed the Bayesian differential privacy, considered the distribution of data and provided a more practical privacy guarantee. Laur et al. [19] proposed a privacy preserving scheme of support vector machine based on secure multiparty computation. For each training or testing phase, their scheme involves multiple parties holding encrypted data and secret sharing obtained

during training. Based on additive homomorphic encryption, Mandal and Gong [20] designed a privacy preserving scheme that performs gradient descent on data owners and cloud server. They achieved secure linear and logistic regression model training. Shen et al. [21] used blockchain technology to establish a secure and reliable data sharing platform among multiple data providers and constructed a privacy preserving support vector machine training scheme based on the Paillier cryptosystem. However, in their scheme, the data provider needs to interact with the cloud server to complete the computation. The computation cost of the data provider is large. Liu et al. [22] proposed a privacy preserving clinical decision support system using the naive Bayes (NB) classifier. The BGV homomorphic encryption system significantly improved the performance. In work [23], a framework for securely and efficiently outsourcing decision tree inference was proposed. Tan et al. [24] proposed a system for privacy-preserving machine learning that implements all operations on the GPU, which makes full use of the computing power of GPU. Zheng et al. [25] combined random permutation and arithmetic secret sharing by the compute-after-permutation technique and built a privacy-preserving machine learning framework. Li et al. [26] proposed a verifiable privacy-preserving machine learning prediction scheme for the edge-enhanced HCPSs, which outputs the verifiable prediction results for users without privacy leakage. Ma et al. [27] designed a lightweight privacy-preserving medical diagnosis mechanism on edge called LPME.

Among them, the SVM algorithm is a research hotspot and has been widely used in different data mining and machine learning schemes. Most of the existing privacy preserving SVM schemes are based on three main privacy preserving technologies: differential privacy (DP), secure multi-party computation (SMC), and homomorphic encryption (HE). DP can significantly improve the calculation and communication efficiency, but the cost is to sacrifice the accuracy of the model by adding random noise [28, 29]. Zhang et al. [30] proposed a general differential privacy model fitting method based on the genetic algorithm, but it reduces the decision accuracy of the model. SMC alleviates the limitation of computing but requires more interaction between participants. This leads to expensive communication overhead [31, 32]. Yu et al. [33] first proposed a privacy preserving SVM classification method based on vertically segmented data. They use SMC technology to obtain the global model, so as to protect the local privacy data and hide the classification model. However, this method requires at least three parties to participate in the calculation, which is complex and inefficient. HE can directly calculate the encrypted data, but it also requires a lot of computing costs [34, 35]. Bajard et al. [36] uses HE technology to protect the decision model and medical data, but it needs high computational load. Therefore, it is necessary to design an efficient and secure SVM scheme for cloud online disease diagnosis service. Wang et al. [37] proposed an efficient privacy preserving outsourced SVM scheme for Internet of medical things deployment, which protected training data privacy and guaranteed the security of the trained SVM model.

In this paper, we propose a new privacy preserving scheme for training and disease diagnosis of the multiclass SVM algorithm. We make a comparison analysis with the schemes in [38–40]. The experimental results demonstrate that our scheme has more practical application values.

3. Preliminaries

In this section, we describe some techniques as the basis of our scheme, including the Paillier cryptosystem, BFV cryptosystem, and SVM algorithm.

3.1. Paillier Cryptosystem. In the training phase, the data are encrypted by the Paillier cryptosystem [41]. The Paillier cryptosystem is a public key cryptosystem with additive homomorphic operation. We will introduce the Paillier cryptosystem as follows.

(i) Key generation: Set the security parameter k. Choose two big primes p, q, $|p| = |q| = k$, $n = p \cdot q$, $\lambda = \mathrm{lcm}(p-1, q-1)$, λ is the Carmichael function of n. Choose a random number $g \in Z_{n^2}^*$, and $\gcd(L(g^\lambda \bmod n^2), n) = 1$, $L(x) = (x-1)/n$. The public key is $pk = (n, g)$. The private key is $sk = \lambda$.

(ii) Encryption: Given $m \in Z_n$. The message m will be encrypted with pk. The ciphertext is expressed as $c = E_{pk}(m) = g^m r^n \bmod n^2$, where $r \in Z_n^*$ is a random number.

(iii) Decryption: According to the key generation stage and Carmichael's theorem, $g^\lambda \equiv 1 \bmod n$. So $g^\lambda = kn + 1$. Then, $m = D_{sk}(c) = (L(c^\lambda \bmod n^2) / L(g^\lambda \bmod n^2)) \bmod n$.

(iv) Homomorphic computation: Given two ciphertexts $E_{pk}(m_1), E_{pk}(m_2)$ under the same public key pk. The homomorphic computations are defined as $E_{pk}(m_1 + m_2) = E_{pk}(m_1) \cdot E_{pk}(m2)$, $E_{pk}(m_1 \cdot m_2) = E_{pk}(m_1)^{m_2}$.

3.2. BFV Cryptosystem. In the prediction phase, the data are encrypted by the BFV cryptosystem [34]. BFV cryptosystem is a leveled-FHE public key cryptosystem based on RLWE, which can support unlimited times additive homomorphic operation and limited times multiplicative homomorphic operation.

(i) Key generation: Generate a polynomial $s = Z[x]/(x^d + 1)$. The private key is defined as $sk = s$. Then, generate a polynomial from ciphertext polynomial space (polynomial s), $a = Z_q[x]/(x^d + 1)$. The polynomial a is used to generate public key. Define a noise polynomial $e \leftarrow \chi$. The notation χ expresses the Gaussian distribution. The public key is $pk = ([-(a \cdot s + e)]_q, a)$.

(ii) Encryption: The message $m \in R_t$. Define $p_0 = pk[0]$, $p_1 = pk[1]$, $u \leftarrow \chi$, $e_1 \leftarrow \chi$, $e_2 \leftarrow \chi$. The ciphertext c is computed as $c = (p_0 \cdot u + t \cdot e_2 + m, p_1 \cdot u + t \cdot e_1)$.

(iii) Decryption: To decrypt the ciphertext c, define $c_0 = p_0 \cdot u + t \cdot e_2 + m, c_1 = p_1 \cdot u + t \cdot e_1$. The message m is computed as $m = (c_0 + c_1 \cdot s) \bmod t$.

(iv) Homomorphic computation: BFV cryptosystem supports ciphertext batch processing. Define two z-dimensional vectors encrypted under public key pk, $E_{pk}(x_1, x_2, \ldots, x_z), E_{pk}(y_1, y_2, \ldots, y_z)$. The homomorphic computations are defined as follows:

$$\begin{aligned} &E_{pk}(x_1 + y_1, x_2 + y_2, \ldots, x_z + y_z) = E_{pk}(x_1, x_2, \ldots, x_z) \\ &\quad + E_{pk}(y_1, y_2, \ldots, y_z), E_pk(x_1 + \cdot y_1, x_2 \cdot y_2, \ldots, x_z \cdot y_z) \\ &= E_{pk}(x_1, x_2, \ldots, x_z) \cdot E_{pk}(y_1, y_2, \ldots, y_z). \end{aligned} \tag{1}$$

3.3. SVM Algorithm. SVM is a classical supervised learning algorithm to solve two kinds of classification problems. The SVM algorithm will find the best hyperplane. The classifier is a decision function $f(X) = \langle W \cdot X \rangle + b$, $f(X) \geq 0$ expresses positive class and $f(X) < 0$ expresses negative class.

There are two training methods for the SVM model: one is based on the SMO algorithm and the other is based on the gradient descent algorithm. Because the operation steps of the SMO algorithm are more complex, which makes a lot computation costs when using encrypted data. Therefore, we choose gradient descent to realize the privacy preserving SVM model training. In the SVM model training process based on the gradient descent method, the objective function $L(X) = (1/2)|W|^2 + C\sum_{i=1}^{n} \max(0, 1 - y_i(\langle W \cdot X \rangle + b)) = (1/2)|W|^2 + C\sum_{i=1}^{n}$ loss needs to be minimized. When $y_i(\langle W \cdot X \rangle + b) \geq 1$, it means that the classification is correct. The $loss = 0$ and the parameters do not need to be updated. When $y_i(\langle W \cdot X \rangle + b) < 1$, it means that the classification is incorrect. The loss $= 1 - y_i(\langle W \cdot X \rangle + b)$ and the parameters need to be updated.

4. System Overview

In this section, we will introduce our system model, security goals, and threat model.

4.1. System Model. Our system model should achieve the privacy preserving training and online disease diagnosis process. Therefore, our system model is designed as shown in Figure 2.

There are six participants in our system model, which are trusted authority (TA), medical centers (MCs), cloud storage server (CSS), cloud computation server (CCS), diagnosis service provider (DSP), and users.

(i) Trusted authority (TA): TA is the fully trusted party of the whole system, which is used to generate and distribute keys for other participants in the system. After initialization, TA will stay offline.

(ii) Medical centers (MCs): Each MC has its own local medical data. To reduce the local storage cost, MCs will outsource the medical data to CSS for storage.

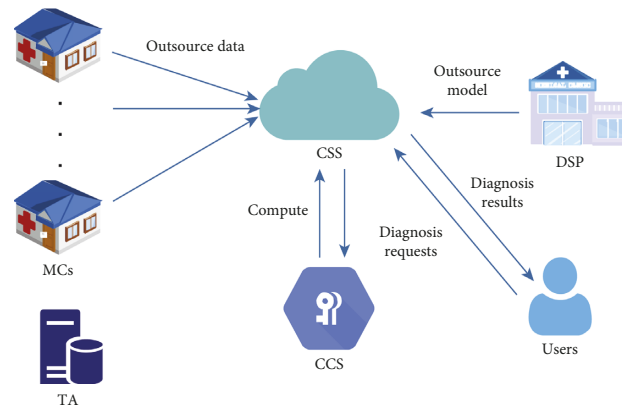

Figure 2: System model.

(iii) Cloud storage server (CSS): CSS has the ability to store and manage outsourced data. CSS can perform privacy preserving computation with its powerful computation power.

(iv) Cloud computation server (CCS): CCS assists CSS to complete privacy preserving computation.

(v) Diagnosis service provider (DSP): DSP wants to train a machine learning model on the outsourced data from MCs and provides online aided disease diagnosis for users. Due to the limited computation and communication ability, DSP will outsource the training and diagnosis to CSS.

(vi) Users: Users are patients or doctors who have unlabeled samples and want to get the diagnosis results. The users will send encrypted diagnosis requests to CSS and obtain the encrypted results. The users can decrypt the results with own private key.

4.2. Security Goals. In order to meet the security requirements of outsourced training and diagnosis, our scheme will achieve the following security goals.

(i) Medical data privacy: The outsourced data of MCs will not be leaked to other participants in the whole machine learning process.

(ii) Model privacy: Other participants cannot learn any useful information about the model of DSP.

(iii) Users privacy: The diagnosis requests and results of users will not be acquired by other participants.

(iv) Intermediate results privacy: In the execution of protocols, any participant will not infer other participants' sensitive information through the intermediate results.

In our scheme, the training and diagnosis processes are completed by CSS and CCS. All participants are semi-honest (or honest-but-curious). Specifically, they will honestly implement the secure computation protocols, but they will try to analyze the sensitive data and intermediate results to infer the useful information of other participants. Like the

previous works, we assume that CSS and CCS will not collude. Because CSS and CCS belong to different commercial companies, they will not collude with each other for their own reputation.

4.3. Threat Model. In this paper, we will define three attacks in our system model.

(i) Eavesdropping attack: This attack means that an adversary can eavesdrop and analyze data during the data transmission. The data transmission includes outsourcing process and the interaction between participants in protocol implementation.

(ii) Honest-but-curious attack: All participants will implement the protocol honestly, but they will infer the useful information during the execution of protocols.

(iii) Client-collusion attack: In the training and diagnosis process, some clients may collude to analyze the useful information of other participants.

5. Proposed Scheme

In this section, we describe the proposed scheme in detail. Our scheme mainly includes system initialization, privacy preserving machine learning training, and online disease diagnosis.

In order to accurately describe our proposed scheme, we give the description of used notations in Table 1.

5.1. System Initialization. In the system initialization phase, TA generates system parameters and distributes the parameters for MCs, CSS, CCS, and DSP, respectively. TA sends the parameters through the secure communication channel. Then, TA will stay offline. We assume that there are m MCs in our system. Because the Paillier cryptosystem and BFV cryptosystem can only encrypt integers, the floating point numbers and negative numbers should be converted into integers. Therefore, all participants should make data conversion before encrypting their sensitive information.

5.1.1. Generate System Parameters

(1) Generate a public-private key pair $(PK_P = (N_P, g), SK_P)$ of the Paillier cryptosystem and a public-private key pair (PK_B, SK_B) of the BFV cryptosystem. The BFV plaintext space is N_B. The public keys are public and the private keys are sent to the CCS.

(2) Generate a public-private key pair (PK_P^C, SK_P^C) of the Paillier cryptosystem and a public-private key pair (PK_B^C, SK_B^C) of the BFV cryptosystem for CSS. The BFV plaintext space is N_B^C. The public keys are public and the private keys are sent to CSS.

(3) Generate a public-private key pair (PK_P^D, SK_P^D) of the Paillier cryptosystem for DSP. The public key is public and the private key is sent to DSP.

Table 1: Notation and definition.

Notation	Definition
$l(x)$	The key length of x
(PK_P, SK_P)	Paillier public-private key pair of CCS
(PK_B, SK_B)	BFV public-private key pair of CCS
(PK_P^C, SK_P^C)	Paillier public-private key pair of CSS
(PK_B^C, SK_B^C)	BFV public-private key pair of CSS
(PK_P^D, SK_P^D)	Paillier public-private key pair of DSP
id_i	The authentication of MC_i
E	The precision of floating point numbers
a_i	The private key of MC_i
$[x]_{PK}$	The ciphertext of x under PK
L	Classification numbers
d	The degree of polynomial

(4) Generate a random integer $\omega \in N_P$. TA randomly splits ω to m integers, satisfying $\omega_1 + \omega_2 + \cdots + \omega_m = \omega$ and sends ω_i to MC_i. Then, generate two lists H and H'. Each list has m random integers, $H = (n_1, n_2, \ldots, n_m)$, $n_i \in N_P$, $H' = (n_1', n_2', \ldots, n_m')$, $n_i' \in N_P$. Each element in H and H' represents the ID of each MC. When MC_i sends authentication id_i to CSS, MC_i will hide g^{a_i} and ω_i with n_i and n_i', respectively. The (n_i, n_i') is sent to MC_i. H and H' are sent to CSS.

5.1.2. Data Conversion. In the machine learning application scenario, data and model parameters contain floating point numbers and negative numbers.

For a floating point number x, we enlarge x to $x \cdot 2^E$ (E is the precision of floating point numbers). For example, given a floating point number $x = 3.61$ and the precision $E = 20$, we can convert x into an integer $x' = 3785359$. For a negative number y, we divide the plaintext space N (N is expressed the plaintext space of the Paillier or BFV cryptosystem) into two parts because all variables and intermediate results in the process of training and prediction are much smaller than $N/2$. An integer in $[0, N/2)$ represents a positive integer and $(N/2, N-1]$ represents a negative integer. When encrypting the negative integer y, it is converted to encrypt $N - y$. If y is both a floating point number and a negative number, y is first converted into a negative integer.

5.2. Privacy Preserving Machine Learning Training. The privacy preserving machine learning training process is completed by CSS and CCS. We assume that the amount of outsourced data is n.

5.2.1. Local Data Outsourcing. To protect the privacy of MCs' local data, MCs will encrypt the data before outsourcing. The outsourcing process of MC_i $(i = 1, \ldots, m)$ is as follows.

(1) MC_i generates a random integer $a_i \in Z_{N_P}$. Computing $pk_i = g^{a_i} \bmod N_P^2$ as public key and the private key is $sk_i = a_i$.

(2) Computing $h_i = g^{a_i+\omega} \bmod N_P^2$.

(3) For each plaintext data, such as x, MC_i will make a data conversion as mentioned in Section 5.1. Then, compute $[x]_{MC_i} = g^x r^{N_P} + h_i$ to encrypt and outsource the encrypted data to CSS for storage.

5.2.2. Secure Basic Building Blocks for Training. To complete the privacy preserving outsourced training, we construct some algorithms as basic building blocks based on the Paillier cryptosystem: secure data aggregation (Block_1), secure multiplication algorithm (Block_2), secure inner product algorithm (Block_3), secure scalar multiplication of vector algorithm (Block_4), and secure symbol judgment algorithm (Block_5). The algorithms will be executed with CSS and CCS.

(1) Secure data aggregation algorithm (Block_1). CSS needs to aggregate MCs' outsourced data before starting machine learning training. The algorithm works as follows and is described in Algorithm 1.

(1) CSS sends a training request to MC_i, $i = 1, 2, \ldots, m$.

(2) After receiving the training request, MC_i computes $id_i = (pk_i + n_i, \omega_i + n_i')$ as authentication (The id_i indicates that CSS is allowed to use the outsourced data of MC_i for training) and sends to CSS.

(3) CSS obtains the pk_i, ω_i of MC_i through the id_i and computes $\omega = \omega_1 + \omega_2 + \cdots + \omega_m$. It should be noted that ω can be obtained only after all MCs have sent their authentication. Then, CSS computes $h_i = g^{a_i+\omega} \bmod N_P^2$ and completes the aggregation.

(2) Secure multiplication algorithm (Block_2). Given two encrypted integers $[x]_{PK_P}$ and $[y]_{PK_P}$, the algorithm needs to compute $[x \cdot y]_{PK_P}$. The algorithm works as follows and is described in Algorithm 2.

(1) CSS generates two random integers R_1, R_2 and $R_1, R_2 \in Z_{N_P}$. Then, it computes by applying the additive homomorphism, obtaining the following results.

$$\begin{aligned} [x + R_1]_{PK_P} &= [x]_{PK_P} \cdot g^{R_1}, \\ [y + R_2]_{PK_P} &= [y]_{PK_P} \cdot g^{R_2}, \end{aligned} \tag{2}$$

Then, sending them to CCS.

(2) CCS generates a random integer $T, T \in Z_{N_P}$. It decrypts $[y + R_2]_{PK_P}$ by using SK_P. Then, it encrypts $(y + R_2 + T) \bmod N_P^C$ with PK_P^C to get $[y + R_2 + T]_{PK_P^C}$. Computing $[xT + R_1T]_{PK_P} = [x + R_1]_{PK_P}^T$ and encrypting T with PK_P. Sending $[y + R_2 + T]_{PK_P^C}$, $[xT + R_1T]_{PK_P}$ and $[T]_{PK_P}$ to CSS.

(3) CSS decrypts $[y + R_2 + T]_{PK_P^C}$ with SK_P^C and computes $y + T$. Then, computing by applying the additive homomorphism, obtaining the following results.

$$\begin{aligned} [xy + xT]_{PK_P} &= [x]_{PK_P}^{y+T}, \\ [xy - R_1T]_{PK_P} &= [xy + xT]_{PK_P} \cdot [xT + R_1T]_{PK_P}^{-1}, \\ [R_1T]_{PK_P} &= [T]_{PK_P}^{R_1}. \end{aligned} \tag{3}$$

Computing the result,

$$[xy]_{PK_P} = [xy - R_1T]_{PK_P} \cdot [R_1T]_{PK_P}. \tag{4}$$

(3) Secure inner product algorithm (Block_3). Given two encrypted vectors $[X]_{PK_P}$, $[Y]_{PK_P}$. The algorithm will compute $[X \cdot Y]_{PK_P}$ and is described in Algorithm 3.

(4) Secure scalar multiplication of vector algorithm (Block_4). Given a encrypted vector $[X]_{PK_P}$ and a encrypted integer $[y]_{PK_P}$, the algorithm will compute $[y \cdot X]_{PK_P}$ and is described in Algorithm 4.

(5) Secure symbol judgment algorithm (Block_5). Given an encrypted integer $[x]_{PK_P}$, the algorithm will compute the sign of $[x]_{PK_P}$. Let judge = 1 if $x \geq 0$ else judge = 0. The algorithm works as follows and is described in Algorithm 5.

(1) CSS chooses a random integer r, $l(r) < l(N_P)/2$. Then, it computes $[x \cdot r]_{PK_P} = [x]_{PK_P}^r$ by applying the additive homomorphism and sends $[x \cdot r]_{PK_P}$ to CCS.

(2) CCS decrypts $[x \cdot r]_{PK_P}$. Let judge = 1 if $x \cdot r \geq 0$ else judge = 0. Then, it sends $[\text{judge}]_{PK_P^C}$ to CSS.

(3) CSS decrypts and obtains the symbol judge.

5.2.3. Privacy Preserving Outsourced Training with Multiclass SVM. In this section, we construct a privacy preserving outsourced training protocol to train a multiclass SVM model using the proposed building blocks. DSP outsources the training task to CSS and CSS completes the aggregation of outsourced data. Then, CSS and CCS complete the model training. After finishing the training, CSS transforms $([W_1]_{PK_P}, \ldots [W_L]_{PK_P})$ into $([W_1]_{PK_P^D}, \ldots [W_L]_{PK_P^D})$. To achieve the transformation, we use the algorithm proposed in reference [38].

For multiclass SVM training, there are two methods: one to rest (ovr) and one to one (ovo). In order to improve the efficiency and reduce the number of iterations, we choose the ovr method for training. We need to construct L binary SVM classifiers, each of which corresponds to one classification. The process is described in Algorithm 6.

5.3. Privacy Preserving Online-Aided Disease Diagnosis. In this section, our proposed scheme consists of four steps: diagnosis outsourcing, secret diagnosis request generation, diagnosis values computation, and diagnosis result generation. The privacy preserving online-aided disease diagnosis is completed by CSS and CCS.

5.3.1. Diagnosis Outsourcing. To reduce the computation and communication overhead, DSP outsources the SVM

Input: the authentication and outsourced data of MC_i.
Output: the training data.
CSS:
(1) Send a training request to MC_i, $i = 1, 2, \ldots, m$.
MCs:
(2) for $i = 1 \longrightarrow m$:
MC_i sends $id_i = (pk_i + n_i, \omega_i + n_i')$ to CSS
end for CCS:
(3) CSS obtains (pk_i, ω_i), $i = 1, 2, \ldots, m$
(4) Compute $\omega = \omega_1 + \omega_2 + \cdots + \omega_m$
(5) for $i = 1 \longrightarrow m$:
Compute $h_i = g^{a_i+\omega} \bmod N_P^2$
For each outsourced data of MC_i, such as $[x]_{MC_i}$,
Compute $[x]_{PK_P} = [x]_{MC_i} - h_i$ to complete aggregate
end for

Algorithm 1: Secure data aggregation (Block_1).

Input: $[x]_{PK_P}, [y]_{PK_P}$
Output: $[xy]_{PK_P}$
CSS:
(1) $R_1, \quad R_2 \in Z_{N_P}$
(2) $[x + R_1]_{PK_P} = [x]_{PK_1} \cdot g^{R_1}$
$[y + R_2]_{PK_P} = [y]_{PK_P} \cdot g^{R_2}$
(3) Send $[x + R_1]_{PK_P}, [y + R_2]_{PK_P}$ to CCS.
CCS:
(4) $T \in Z_{N_P}$
(5) Decrypt $[y + R_2]_{PK_P}$
(6) $[xT + R_1T]_{PK_P} = [x + R_1]_{PK_P}^T$
(7) Encrypt $(y + R_2 + T) \bmod N_P^C$ with PK_P^C
(8) Encrypt T with PK_P
(9) Send $[xT + R_1T]_{PK_P}, [y + R_2 + T]_{PK_P^C}$ and $[T]_{PK_P}$ to CSS.
CSS:
(10) Decrypt $[y + R_2 + T]_{PK_P^C}$ with SK_P^C and Compute $y + T$
(11) $[xy + xT]_{PK_P} = [x]_{PK_P}^{y+T}$
(12) $[xy - R_1T]_{PK_P} = [xy + xT]_{PK_P} \cdot [xT + R_1T]_{PK_P}^{-1}$
(13) $[R_1T]_{PK_P} = [T]_{PK_P}^{R_1}$
(14) $[xy]_{PK_P} = [xy - R_1T]_{PK_P} \cdot [R_1T]_{PK_P}$

Algorithm 2: Secure multiplication (Block_2).

Input: $[X]_{PK_P}, [Y]_{PK_P}$
Output: $[X \cdot Y]_{PK_P}$
CSS:
(1) Define $[X \cdot Y]_{PK_P} = [1]_{PK_P}$.
(2) for $i = 1 \longrightarrow X$.length:
$[X \cdot Y]_{PK_P} \cdot = \text{Block\_1}([x_i]_{PK_P}, [y_i]_{PK_P})$
end for

Algorithm 3: Secure inner product (Block_3).

model parameters to CSS and authorizes CSS to provide diagnosis service for users.

The SVM parameters of DSP are expressed as $(W^1, W^2, \ldots, W^L)$ (There are L classifiers),

$$W^i = \left(w_1^i, w_2^i, \ldots, w_{t+1}^i\right), \quad (i = 1, 2, \ldots, L). \tag{5}$$

Input: $[X]_{PK_P}$, $[y]_{PK_P}$
Output: $[y \cdot X]_{PK_P}$
CSS:
(1) Define $[y \cdot X]_{PK_P} = [1, 1, \ldots, 1]_{PK_P}$.
(2) for $i = 1 \longrightarrow X$.length:
$[yx_i]_{PK_P} = \text{Block\_1}([y]_{PK_P}, [x_i]_{PK_P})$
end for

Algorithm 4: Secure scalar multiplication of vector (Block_4).

Input: $[x]_{PK_P}$
Output: judge
CSS:
(1) Choose a random integer r, $l(r) < l(N_P)/2$
(2) $[x \cdot r]_{PK_P} = [x]^r_{PK_P}$
(3) Send $[x \cdot r]_{PK_P}$ to CCS
CCS:
(4) Decrypt $[x \cdot r]_{PK_P}$
(5) if $x \cdot r \geq 0$: judge = 1, else: judge = 0
(6) Encrypt judge with PK_P^C
(7) Send $[\text{judge}]_{PK_P^C}$ to CSS

Algorithm 5: Secure symbol judgment (Block_5).

Input: outsourced data of MCs
$([X_1]_{PK_P}, [y_1]_{PK_P}), \ldots, ([X_n]_{PK_P}, [y_n]_{PK_P})$,
iterations T, learning rate learnrate,
regularization parameter z
Output: L encrypted binary SVM classifiers parameters
$([W_1]_{PK_P}, \ldots [W_L]_{PK_P})$
(1) for $k = 1 \longrightarrow L$:
for it $= 1 \longrightarrow T$:
$[\text{grad}]_{PK_P} = [W_k]_{PK_P}$
for $i = 1 \longrightarrow n$:
(2) $[W_k \cdot X_i]_{PK_P} = \text{Block\_3}([W_k]_{PK_P}, [X_i]_{PK_P})$
(3) $\text{tmp} = \text{Block\_2}([y_i]_{PK_P}, [W_k \cdot X_i]_{PK_P}) \cdot [1]^{-1}_{PK_P}$
(4) if Block_5(tmp) == 0:
$[y_i \cdot X_i]_{PK_P} = \text{Block\_4}(y_i, X_i)$
$[\text{grad}]_{PK_P} \cdot = [W_j] \cdot [y_i \cdot X_i]^{z \cdot (N_P - 1)}_{PK_P}$
end for
(5) $[W]_{PK_P} \cdot = [\text{grad}]^{\text{learnrate} \cdot (N_P - 1)}_{PK_P}$
end for
end for
(6) return $([W_1]_{PK_P}, \ldots, [W_L]_{PK_P})$

Algorithm 6: Secure multiclass SVM training.

For W^i and the corresponding class result class^i, DSP generates a $t+1$-dimensional random integer vector $R^i = (R^i_1, R^i_2, \ldots, R^i_{t+1})$ and a random integer r^i, $l(R^i_j) = l(r^i) < l(N^C_B)/2$, $l(R^i_j) < l(N_B)/2$. Then, DSP computes $W^i + R^i$ and class$^i + r^i$ to hide the parameters class results.

According to the combination of subtraction of L random integer vectors, DSP constructs a combination table. The combination table has C^2_L values, as shown in Table 2. The values in combination table are used to eliminate the blinding factors in subsequent computation.

Table 2: Combination table.

$(1,2)$: $index = 1 \cdot L + 2$	...	$(L-1, L)$: $index = (L-1) \cdot L + L$
$R^1 - R^2$	...	$R^{L-1} - R^L$

DSP encrypts $W^i + R^i$ with PK_B, $\text{class}^i + r^i$ with PK_B^C, r^i with PK_P and all values of combination table with PK_B^C. Then, DSP sends them as the outsourced parameters to CSS. After receiving the outsourced parameters, CSS decrypts $[\text{class}^i + r^i]_{PK_B^C}$ and the combination table with SK_B^C. CSS computes as follows:

$$\left[\text{class}^i\right]_{PK_P} = \left[\text{class}^i + r^i_{PK_P}\right] \cdot \left[r^i\right]^{-1}_{PK_P}, \quad i = 1, 2, \ldots, L. \tag{6}$$

5.3.2. Secret Diagnosis Request Generation. For user_i, the symptom is expressed as $X^i = (x_1^i, x_2^i, \ldots, x_t^i, 1)$ (The last 1 is added to facilitate the computation of vector inner product). The user_i generates a $t+1$-dimensional random integer vector $T^i = (T_1^i, T_2^i, \ldots, T_j^i, \ldots, T_{t+1}^i)$ and $l(T_j^i) < l(N_B^C)/2$, $l(T_j^i) < l(N_B)/2$. Then, user_i hides plaintext symptoms $X^i + T^i$.

The user_i encrypts symptom $X^i + T^i$ with PK_B and encrypts T^i with PK_B^C. Let S as the secret prediction request of user_i.

$$S = \left(\left[X^i + T^i\right]_{PK_B}, \left[T^i\right]_{PK_B^C}\right). \tag{7}$$

Then, the user_i sends S to CSS.

5.3.3. Diagnosis Value Computation. In our proposed diagnosis scheme, it is a multiclassification problem, so it is necessary to compute the diagnosis value of each classification. After receiving the secret prediction request S, CSS decrypts $[T^i]_{PK_B^C}$ with SK_B^C. Then, it computes $[X^i + T^i]_{PK_B} - T^i$ by the homomorphic operation of the BFV cryptosystem.

According to the decision function $f(X) = W \cdot X + b$ of the SVM algorithm, a diagnosis value needs to be computed by one multiplication homomorphic operation and one addition homomorphic operation. Because the BFV encryption algorithm supports ciphertext packaging, batch operation can be realized and the computation efficiency is significantly improved. The process is described in Algorithm 7.

5.3.4. Diagnosis Result Generation. After computing the diagnosis values, CSS obtains L encrypted diagnosis values and each value corresponds to a class result. Then, CSS needs to select the classification corresponding to the maximum value from the L encrypted values as the diagnosis result.

Therefore, we design a secure maximum find protocol and a secure comparison algorithm. In this process, CSS and CCS jointly execute the protocol.

(1) Secure maximum finding. CSS sets an initial maximum position pos = 1. Then, CSS executes L cycles and each cycle executes a secure comparison algorithm to continuously update the pos value.

After L cycles, CSS obtains the final diagnosis result $[\text{class}^{\text{pos}}]_{PK_P}$ and converts $[\text{class}^{\text{pos}}]_{PK_P}$ into $[\text{class}^{\text{pos}}]_{PK_{\text{user}_i}}$ under the public key PK_{user_i} of user_i. To achieve the transformation, we use the algorithm proposed in literature [38]. Then, CSS sends $[\text{class}^{\text{pos}}]_{PK_{\text{user}_i}}$ to user_i. The user_i decrypts the encrypted result with SK_{user_i}. The process is described in Algorithm 8.

(2) Secure comparison (SC). For the i-th cycle, CSS computes $[\Delta_{pos-i}] = [(W^{\text{pos}} + R^{\text{pos}})X^i]_{PK_B} - [(W^j + R^j)X^i]_{PK_B}$. Then, according to pos and j, computing index = $\text{pos} \cdot L + j$. The index corresponds to the value $(R^{\text{pos}} - R^j)_{\text{index}}$ in the combination table and computing as follows:

$$\begin{aligned} \left[X^i\left(R^{\text{pos}} - R^j\right)_{\text{index}}\right]_{PK_B} &= \left[X^i\right]_{PK_B} \cdot \left(R^{\text{pos}} - R^j\right)_{\text{index}}, \\ \left[\Delta_{\text{pos}-j}'\right]_{PK_B} &= \left[\Delta_{\text{pos}-j}\right]_{PK_B} - \left[X^i\left(R^{\text{pos}} - R^j\right)_{\text{index}}\right]_{PK_B}. \end{aligned} \tag{8}$$

At this time, $[\Delta_{pos-j}']_{PK_B}$ has eliminated $(R^{\text{pos}} - R^j) \cdot X^i$ in $[\Delta_{\text{pos}-j}]_{PK_B}$.

CSS chooses $t+1$ equal random integers r', $R' = (r', \ldots, r')$ and $l(r') < l(N_B)/2$. Computing $[\Delta_{\text{pos}-j}']_{PK_B} = [\Delta_{\text{pos}-j}']_{PK_B} \cdot R'$. Then, CSS chooses $t+1$ different random integers, $R'' = (r_1'', \ldots, r_{t+1}'')$, $l(r_1'') = \cdots = l(r_{t+1}'') < l(N_B)/2$ and computing $[\Delta_{pos-j}']_{PK_B} = [\Delta_{pos-j}']_{PK_B} + R''$. Summing all elements in R'' to get R_{css} and encrypting it with PK_P. CSS sends $[\Delta_{\text{pos}-j}']_{PK_B}$, $[R_{\text{css}}]_{PK_P}$ to CCS.

CCS decrypts $[\Delta_{\text{pos}-j}']_{PK_B}$ with SK_B and $[R_{\text{css}}]_{PK_P}$ with SK_P, $((w_1^{pos}x_1^i - w_1^j x_1^i)r' + r_1'', \ldots, (w_{t+1}^{pos}\ x_{t+1}^i - w_{t+1}^j x_{t+1}^i)r' + r_{t+1}'')$. Then, summing each dimension, $s = \text{sum}((w_1^{\text{pos}}x_1^i - w_1^j x_1^i)r' + r_1'', \ldots, (w_{t+1}^{\text{pos}}x_{t+1}^i - w_{t+1}^j x_{t+1}^i)r' + r_{t+1}'')$.

CCS removes R_{css} from s by computing $(s - R_{\text{css}}) \bmod N_B$. Let judge = 1 if $s > N_B/2$, else judge = 0.

CCS encrypts judge with PK_B^C and sends it to CSS. CSS decrypts it and if judge = 1, updates the value of pos.

The process is described in Algorithm 9.

6. Security Analysis

In this section, we analyze the security of the proposed scheme. The focus is on the outsourced data of MCs, the SVM model parameters of DSP, the symptoms, and diagnosis results of users.

6.1. Security Analysis of Training. In the training phase, the outsourced data of MCs and the SVM model parameters of DSP need privacy preserving. The training protocol is composed of building blocks designed in Section 5.2.2, which are completed by CSS and CCS. According to the

Input: $[X^i]_{PK_B}$, $[W^j + R^j]_{PK_B}$, $j = 1, 2, \ldots, L$
Output: L encrypted diagnosis values
$[(W^1 + R^1)X^i]_{PK_B}, \ldots, [(W^L + R^L)X^i]_{PK_B}$
CSS:
(1) for $j = 1 \longrightarrow L$:
$[(W^j + R^j)X^i]_{PK_B} = [W^j + R^j]_{PK_B} \cdot [X^i]_{PK_B}$
end for

Algorithm 7: Diagnosis value computation.

Input: L diagnosis values and corresponding class results
$[(W^j + R^j)X^i]_{PK_B}, \text{class}^j, \quad j = 1, 2, \ldots, L;$
initial pos = 1
Output: $[\text{class}^{\text{pos}}]_{PK_{\text{user}_i}}$
CSS:
for $j = 2 \longrightarrow L$:
(1) judge = $SC([(W^{\text{pos}} + R^{\text{pos}})X^i]_{PK_B}, [(W^j + R^j)X^i]_{PK_B})$
(2) if judge = 1:
pos = i
end for
(3) Transform $[\text{class}^{\text{pos}}]_{PK_p}$ into $[\text{class}^{\text{pos}}]_{PK_{\text{user}_i}}$ with CCS
(4) Send $[\text{class}^{\text{pos}}]_{PK_{\text{user}_i}}$ to user_i.
user_i:
(5) user_i decrypts $[\text{class}^{\text{pos}}]_{PK_{\text{user}_i}}$ with SK_{user_i}.

Algorithm 8: Secure maximum finding.

Input: $[(W^{\text{pos}} + R^{\text{pos}})X^i]_{PK_B}$, $[(W^j + R^j)X^i]_{PK_B}$
Output: judge
CSS:
(1) $[\Delta_{\text{pos}-j}]_{PK_B} =$
$[(W^{\text{pos}} + R^{\text{pos}})X^i]_{PK_B} - [(W^j + R^j)X^i]_{PK_B}$
(2) index = pos $\cdot L + j$
(3) $[X^i(R^{\text{pos}} - R^j)_{\text{index}}]_{PK_B} = [X^i]_{PK_B} \cdot (R^{\text{pos}} - R^j)_{\text{index}}$
(4) $[\Delta_{\text{pos}-j}']_{PK_B} = [\Delta_{\text{pos}-j}]_{PK_B} - [X^i(R^{\text{pos}} - R^j)_{\text{index}}]_{PK_B}$
(5) Generate $R\prime$.
(6) $[\Delta_{\text{pos}-j}']_{PK_B} = [\Delta_{\text{pos}-j}']_{PK_B} \cdot R\prime$
(7) Generate $R\prime\prime = (r_1'', r_2'', \ldots, r_{t+1}'')$.
(8) $[\Delta_{\text{pos}-j}']_{PK_B} = [\Delta_{\text{pos}-j}']_{PK_B} + R\prime\prime$
(9) $R_{\text{css}} = r_1' + r_2' + \cdots + r_{t+1}'$
(10) Send $[\Delta_{\text{pos}-i}']_{PK_B}, [R_{\text{css}}]_{PK_p}$ to CCS.
CCS:
(11) Decrypt $[\Delta_{\text{pos}-i}']_{PK_B}, [R_{\text{css}}]_{PK_p}$.
(12) $s = \text{sum}((w_1^{\text{pos}} x_1^i - w_1^j x_1^i)r\prime + r_1'', \ldots,$
$(w_{t+1}^{\text{pos}} x_{t+1}^i - w_{t+1}^j x_{t+1}^i)r\prime + r_{t+1}'')$
(13) $s = (s - R_{\text{css}}) \bmod N_B$
(14) if $s > N_B/2$: judge = 1
else: judge = 0
(15) Encrypt judge. Send it to CSS
CSS:
(16) Decrypt $[\text{judge}]_{PK_B^C}$

Algorithm 9: Secure comparison (SC).

threat models proposed in Section 4.3, we analyze the security of the training protocol.

6.1.1. Eavesdropping Attack. The data transmission process in the training phase includes that MCs outsources the encrypted data to CSS and the interactions of training protocol between CSS and CCS.

In the outsourcing process, the data of MC_i have been encrypted. MC_i combines the system public key PK_P, parameter ω, and its own public key g^{a_i} to ensure that the data are hidden while encrypting. Suppose an adversary obtains the private key SK_P and eavesdrops when MCs outsource their data to CSS. Because the data of MC_i have been encrypted, such as $[x]_{MC_i} = g^x r^{N_P} + h_i$, the adversary cannot obtain any useful information. Similarly, the authentications are also hidden by random numbers. In the training protocol execution process, CSS and CCS will interact and the transformed data have been encrypted and hidden the real values with random numbers. The adversary also cannot obtain any useful information.

6.1.2. Honest-But-Curious Attack. During the training phase, CSS and CCS will get some intermediate results from the proposed building blocks in Section 5.2.2.

In the Block_2, CSS hides x, y with R_1, R_2 by homomorphic operation before sending them to CCS. Then, CCS sends $[xT + R_1T]_{PK_P}$, $[y + R_2 + T]_{PK_P^C}$ and $[T]_{PK_P}$ to CSS after computing. Therefore, both CSS and CCS cannot learn any useful information about x, y. Because the Block_3 and Block_4 are designed based on the Block_2, we will not analyze them. In the Block_5, CSS hides x with r and sending $[x \cdot r]_{PK_P}$ to CCS. CCS can only know the symbol of x, but cannot obtains the real value of x. CCS only returns the result judge (0 or 1) to CSS. Through the abovementioned analysis, CSS and CCS cannot learn any useful information in the training process.

6.1.3. Client-Collusion Attack. For MCs, each MC_i only know its own ω_i. Therefore, if $(m-1)$ MCs collude with each other to steal the privacy of another MC, they cannot learn any useful information.

6.2. Security Analysis of Disease Diagnosis. In the diagnosis phase, the SVM parameters of DSP, the symptom $\mathbf{X}^i$ and the diagnosis result $\text{class}^{\text{pos}}$ of user_i need privacy preserving. The diagnosis process consists of diagnosis outsourcing, secret diagnosis request generation, diagnosis value computation, and diagnosis result generation. Therefore, we conduct security analysis on the main steps by the threat model.

6.2.1. Eavesdropping Attack. The data transmission process includes that DSP outsources $[W^i + R^i]_{PK_B}$, $[\text{class}^i + r^i]_{PK_B^C}$, $[r^i]_{PK_P}$, $i = 1, 2, \ldots, L$ and $[R^1 - R^2]_{PK_B^C}, \ldots, [R^{L-1} - R^L]_{PK_B^C}$ to CSS, user_i sends request S to CSS and the interaction of diagnosis process between CSS and CCS.

Through the encrypted data of outsourcing process, it can be seen that the adversary(CCS) can only decrypt $[W^i + R^i]_{PK_B}$ and $[r^i]_{PK_P}$ with SK_B and SK_P. However, the adversary cannot learn W^i because of the R^i and the r^i do not contain any useful information. When user_i sends S to CSS, the symptom X^i may be eavesdropped and decrypted by the adversary, but X^i is hidden by random numbers. In the interaction of SC algorithm between CSS and CCS, all transmitted data are hidden by random numbers and ciphertext state, so the adversary cannot learn any useful information.

6.2.2. Honest-But-Curious Attack. In the diagnosis value computation process, CSS can only obtain the L encrypted diagnosis values under PK_B and does not know the corresponding classification meaning. The whole process is executed in the ciphertext state, so CSS cannot learn any useful information. The process of diagnosis result generation consists of secure maximum finding protocol and secure comparison algorithm. When CSS and CCS execute the secure comparison algorithm, CSS computes the difference between the two encrypted vectors to be compared. The obtained difference vector can confuse the positive and negative of the two numbers on each dimension of the original two vectors. At the same time, random integers are used to hide the difference vector. After decrypting the difference vector, CCS can eliminate the random number only after summing. During this process, CSS and CCS cannot obtain any useful information.

After CSS and CCS execute secure maximum finding protocol, CSS obtains the diagnosis result $[\text{class}^{\text{pos}}]_{PK_P}$. When performing key conversion on $[\text{class}^{\text{pos}}]_{PK_P}$, CSP hides $\text{class}^{\text{pos}}$ with a random integer R. Then, sending $[\text{class}^{\text{pos}} + R]_{PK_P}$ to CCS. CCS can decrypt it. However, because there is a random integer hidden, CCS cannot obtain $\text{class}^{\text{pos}}$.

6.2.3. Client-Collusion Attack. For all users, they can only get the diagnosis results and cannot get any other information. Therefore, our proposed scheme can resist the client-collusion attack.

7. Performance Evaluation

In this section, we implemented our scheme and evaluated the performance of training and diagnosis.

Our experimental environment is shown in Table 3.

In our experiments, we evaluated our proposed scheme with a real dataset from UCI machine learning library called dermatology. The dermatology dataset is a multiclassification dataset with 6 categories and 34 symptoms.

7.1. Privacy Preserving Machine Learning Training Evaluation

7.1.1. Effect of Key Length on Computation Overhead. The key length in cryptosystem has a great impact on efficiency and security. Therefore, we tested the data encryption time and main building blocks time (Block_1 and Block_3), which have high computation overhead. The test results are shown in Table 4.

Table 3: Experimental environment.

Operating system	Windows 10
CPU	Intel (R) Core(TM)i7-10510U, 1.80 GHz, 2.30 GHz
Memory	8 G
Program language	C++

Table 4: Computation overhead under different key length.

Key length (bit)	Data encryption (s)	Block_1 (s)	Block_3 (s)
$l = 256$	$4.15e\,10 - 4$	$2.08\,e10 - 5$	0.094
$l = 512$	$2.52e\,10 - 3$	$5.0e\,10 - 5$	0.417
$l = 1024$	0.0153	$2.1e\,10 - 4$	2.52
$l = 2048$	0.103	$9.9e\,10 - 4$	17.2

From Table 4, it can be seen that the increase of key length has a great impact on the computation overhead. Based on the experimental results and security considerations, the key length of the Paillier cryptosystem is set to 1024 bit in the training phase.

7.1.2. Privacy Preserving Multiclass SVM Training Analysis. In order to meet the requirements of data encryption, we convert all floating-point numbers to integers. The conversion accuracy E of floating-point numbers has a great impact on the accuracy of the SVM model. We tested the accuracy of the SVM model under different E values; the results are shown in Figure 3.

Through the abovementioned experimental analysis, it can be seen that the larger the E, the higher the accuracy of the model. With the increase of E, the accuracy of the model tends to be stable. When $E = 20$, the accuracy of the model is the highest. At the same time, we also used the gradient descent method to train the SVM model in the plaintext state. We compared the accuracy with the model trained in ciphertext state and the results are shown in Table 5.

Through the abovementioned experimental analysis, it can be seen that the accuracy of our proposed scheme is the same as the plaintext state (98.61%). Therefore, it is verified that our proposed scheme is correct and available.

7.2. Privacy Preserving Online-Aided Disease Diagnosis Evaluation. We implemented our proposed scheme by using SEAL library in the diagnosis phase.

7.2.1. Noise Effect of BFV Cryptosystem. When using the BFV cryptosystem for homomorphic operation, the influence of noise needs to be considered. The noise of ciphertext will be increased when the multiplication homomorphic operation is carried out. If the noise is too large after computation, the correct result cannot be obtained after decryption.

Therefore, the BFV cryptosystem in SEAL will set the noise budget during initialization. If the noise budget is greater than 0 after the computation, it can be decrypted correctly. The value of noise budget is related to the setting of parameters. We evaluated the influence of poly module degree (d) on the encryption time, the change of noise budget after homomorphic operation, the computation time and whether the decryption result is correct. The results are shown in Table 6. It can be seen that the noise consumption of the BFV cryptosystem is relatively large when performing multiplication homomorphism, so the BFV cryptosystem can only perform multiplication homomorphism for a limited number of times. When computing the diagnosis values, only one inner product operation and one addition operation are required. Therefore, it is completely feasible to use the BFV cryptosystem.

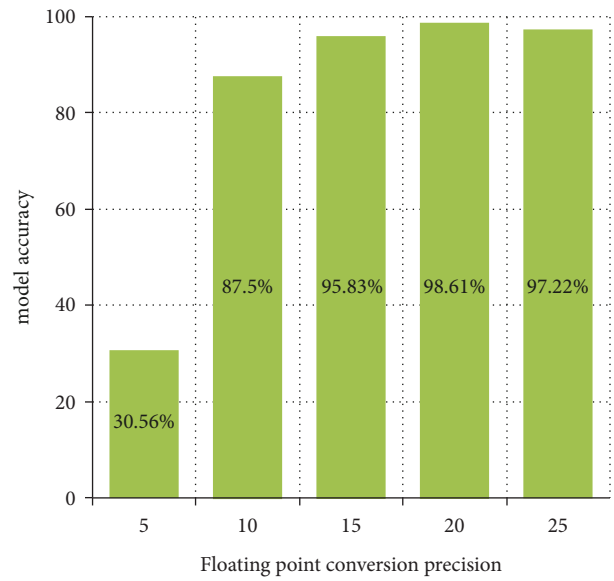

Figure 3: The influence of precision.

We comprehensively consider the encryption time and computation time and ensure that the computation results can be decrypted correctly. The parameter we set is poly module degree (d) = 8192.

7.2.2. Influence of Different Classification Numbers on Computation Overhead. When using the BFV cryptosystem to encrypt data, multiple plaintext data can be packaged and encrypted into a ciphertext. The number of classifications is L.

Table 5: Comparison analysis of model accuracy.

Dataset	Plaintext state	Our proposed scheme
Dermatology	98.61%	98.61%

Table 6: The influence of poly modulus degree (d) on noise budget.

d	Encryption time (s)	Initial noise budget (bit)	Noise budget (bit) after operation	Computation time (s)	Decryption result (correct or wrong)
2048	0.013	2	0	0.012	×
4096	0.023	9	0	0.031	×
8192	0.69	110	64	0.123	✓
32768	1.178	761	713	2.268	✓

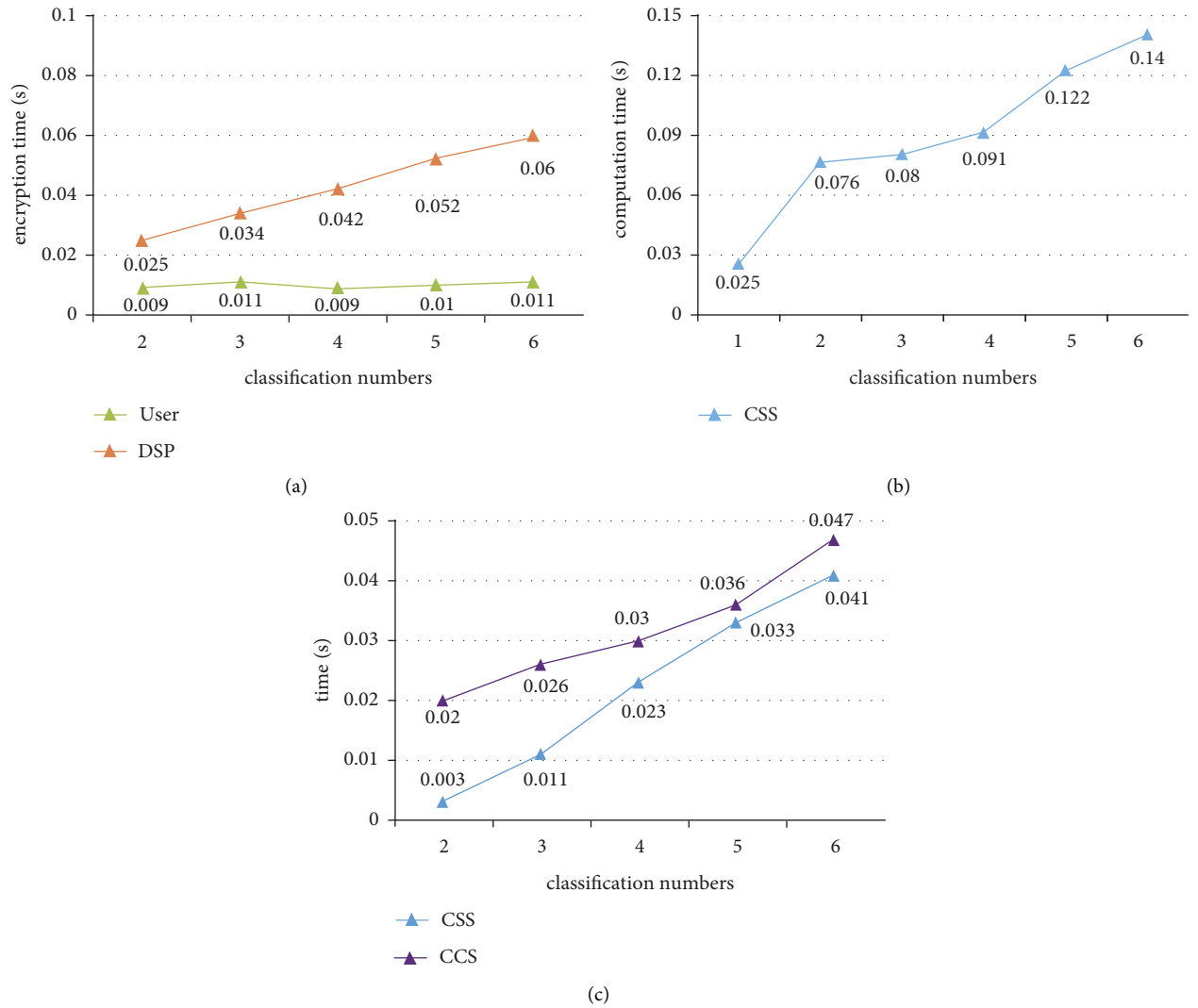

Figure 4: Influence of different classification numbers on computation overhead. (a) Data encryption, (b) diagnosis values computation, and (c) diagnosis result generation.

We tested the impact of different L on user$_i$ and DSP. The results are shown in Figure 4(a). With the increase of L, the encryption time of DSP is gradually increasing, and the encryption time of user$_i$ can be considered as unchanged. We also tested the impact of different L on the diagnosis values computation of CSS. The results are shown in Figure 4(b). With the continuous increase of L, the computation time for CSS is also increasing. The process of

Table 7: Comparison analysis.

Schemes	Data encryption (s)	Diagnosis values computation (s)	Total time (s)
Reference [38], $l = 512$	0.658	0.005	0.663
Reference [39], $l = 512$	0.57	0.552	1.122
Reference [40], $l = 512$	0.004	1.092	1.096
Ours	0.008	0.096	0.104

Table 8: Comparison analysis of the cloud server.

Schemes	Data encryption (s)	Diagnosis value computation (s)	Total time (s)
Reference [38], $l = 512$	0	0	0
Reference [39], $l = 512$	0.57	0.001	0.571
Reference [40], $l = 512$	0	1.092	1.092
Ours	0	0.096	0.096

Table 9: Comparison analysis of the client.

Schemes	Data encryption (s)	Diagnosis value computation (s)	Total time (s)
Reference [38], $l = 512$	0.658	0.005	0.663
Reference [39], $l = 512$	0	0.551	0.551
Reference [40], $l = 512$	0.004	0	0.004
Ours	0.008	0	0.008

Table 10: Comparison analysis of diagnosis result generation.

Schemes	Cloud server computation (s)	Client computation (s)	Diagnosis result generation (s)
Reference [38], $l = 512$	1.584	0.097	1.681
Reference [39], $l = 512$	0.007	0.041	0.048
Reference [40], $l = 512$	0.101	0	0.101
Ours	0.071	0	0.071

generating diagnosis result is jointly completed by CSS and CCS. We tested the effect of different L on the diagnosis result generation. The results are shown in Figure 4(c). With the continuous increase of L, the time for CSS and CCS is also increasing.

7.2.3. Comparison Analysis of Secret Diagnosis Request Generation and Diagnosis Values Computation. In our proposed scheme, secret diagnosis request generation can be regarded as data encryption of user_i and diagnosis value computation can be regarded as homomorphic operation. We compared with the other three privacy preserving schemes. The results are shown in Table 7.

Through the comparison analysis, it can be seen that the time of data encryption in our proposed scheme is significantly reduced compared with [38, 39]. In the computation of decision function, our scheme has significantly reduced the computational cost compared with the scheme in [39, 40]. At the same time, it can be seen from the total time that our proposed scheme is significantly lower than the other three schemes.

Next, we make further analysis. The names of participants may be slightly different in different schemes. In order to facilitate analysis, we divided participants into cloud server and client. We compared the computation overhead of cloud server and client, respectively. The results are shown in Tables 8 and 9.

In our proposed scheme, the client only needs to encrypt the data and can be offline after uploading the data to the cloud server. The cloud server only needs to compute the decision function. This model reduces the computation overhead of the client to the greatest extent and performs privacy preserving computation through the powerful computing power of the cloud server. In scheme [38], the cloud server does not participate in the whole process, so it brings heavy computation overhead to the client. In scheme [39], the computation of the diagnosis values needs to be completed by the cloud server and the client. Therefore, it not only brings heavy computation overhead to the client but also requires the client to always stay online in this process.

7.2.4. Comparison Analysis of Diagnosis Result Generation. In our proposed scheme, after CSS completes the diagnosis values computation, it will jointly execute the secure protocol with CCS to generate the diagnosis result. We continued to make comparison analysis with schemes in [38–40]. The results are shown in Table 10.

Through the comparison analysis in Table 10, it can be seen that the computation time of our proposed scheme is

Table 11: Comprehensive comparison analysis.

	Data encryption (s)	Diagnosis value computation (s)	Diagnosis result generation (s)	Total time (s)
Reference [38], $l = 512$	0.658	0.005	1.681	2.389
Reference [39], $l = 512$	0.57	0.552	0.048	1.17
Reference [40], $l = 512$	0.004	1.092	0.101	1.197
Ours	0.008	0.096	0.071	0.175

Table 12: Scheme summary.

Schemes	Multidata owners	Training	Multiclassification	Low client overhead
Reference [38]	×	×	✓	×
Reference [39]	×	×	✓	×
Reference [40]	×	×	✓	✓
Reference [37]	✓	✓	×	✓
Ours	✓	✓	✓	✓

significantly lower than the scheme in references [38, 40]. In our proposed scheme, the client does not need to participate in the process of diagnosis result generation. The schemes in references [38, 39] require the participation of the client, which brings heavy computation overhead to the client.

7.2.5. Comprehensive Comparison Analysis. We made a comparison analysis of the whole privacy preserving online disease diagnosis process. It is divided into the secret diagnosis request generation (data encryption), diagnosis value computation, and diagnosis result generation. The results are shown in Table 11.

Through the comparison analysis in Table 11, the total time of our proposed scheme is significantly lower than the schemes in references [38–40]. Considering that in the actual application scenario, a large number of users will constantly initiate secret diagnosis requests. It is very important to be able to quickly respond to the diagnosis results for users. Therefore, our scheme has more practical application value. Then, we made a summary as shown in Table 12.

8. Conclusion

In this paper, we propose an efficient and privacy preserving outsourced multiclass SVM training and online-aided disease diagnosis scheme. We design some secure basic operation algorithms for machine learning training over the outsourced data from multiple data owners. We achieve a privacy preserving multiclass SVM training based on the basic operation algorithms. In the diagnosis phase, we achieve a privacy preserving multiclass diagnosis through our proposed the secure maximum find algorithm and secure comparison algorithm. Security analysis proves that our proposed scheme ensures that outsourced data, model parameters, users' symptoms, and diagnosis results will not be leaked. Experimental evaluation illustrates that our proposed scheme significantly reduces the computation overhead. In the future, we will study more efficient and privacy preserving machine learning schemes.

Data Availability

The data supporting the results of this study can be obtained from the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The work was supported in part by the National Natural Science Foundation of China (61862052) and the Science and Technology Foundation of Qinghai Province (2019-ZJ-7065).

References

[1] M. Kim, Y. Song, S. Wang, Y. Xia, and X. Jiang, "Secure logistic regression based on homomorphic encryption: design and evaluation," *JMIR medical informatics*, vol. 6, no. 2, p. e19, 2018.

[2] X. Yang, R. Lu, J. Shao, X. Tang, and H. Yang, "An efficient and privacy-preserving disease risk prediction scheme for e-healthcare," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3284–3297, 2019.

[3] E. Ayday, J. L. Raisaro, P. J. McLaren, J. Fellay, and J.-P. Hubaux, "Privacy-preserving computation of disease risk by using genomic, clinical, and environmental data," in *Proceedings of the 2013 { USENIX} Workshop on Health Information Technologies (HealthTech 13)*, Washington, DC, USA, August 2013.

[4] J. Zhang, L. Zhang, M. He, and S.-M. Yiu, "Privacy-preserving disease risk test based on bloom filters," in *Proceedings of the International Conference on Information and Communications Security*, pp. 472–486, Singapore, December 2017.

[5] 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.

[6] N. Bouguila, "Hybrid generative/discriminative approaches for proportional data modeling and classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 12, pp. 2184–2202, 2012.

[7] M. Ćwiklińska-Jurkowska, "Performance of the support vector machines for medical classification problems,"

Biocybernetics and Biomedical Engineering, vol. 29, no. 4, pp. 63–81, 2009.

[8] S. G. Teo, S. Han, and V. C. Lee, "Privacy preserving support vector machine using non-linear kernels on hadoop mahout," in *Proceedings of the 2013 IEEE 16th International Conference on Computational Science and Engineering*, pp. 941–948, IEEE, Sydney, Australia, December 2013.

[9] M. Z. Omer, H. Gao, and F. Sayed, "Privacy preserving in distributed svm data mining on vertical partitioned data," in *Proceedings of the 2016 3rd International Conference on Soft Computing & Machine Intelligence (ISCMI)*, pp. 84–89, IEEE, Dubai, UAE, November 2016.

[10] L. Wang, J. J. Shi, C. Chen, and S. Zhong, "Privacy-preserving face detection based on linear and nonlinear kernels," *Multimedia Tools and Applications*, vol. 77, no. 6, pp. 7261–7281, 2018.

[11] M. S. Riazi, C. Weinert, O. Tkachenko, E. M. Songhori, T. Schneider, and F. Koushanfar, "Chameleon: a hybrid secure computation framework for machine learning applications," in *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, pp. 707–721, Incheon, Korea, June 2018.

[12] 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, 2017.

[13] Y. Rahulamathavan, R. C.-W. Phan, S. Veluru, K. Cumanan, and M. Rajarajan, "Privacy-preserving multi-class support vector machine for outsourcing the data classification in cloud," *IEEE Transactions on Dependable and Secure Computing*, vol. 11, no. 5, pp. 467–479, 2014.

[14] S. Perveen, M. Shahbaz, K. Keshavjee, and A. Guergachi, "A systematic machine learning based approach for the diagnosis of non-alcoholic fatty liver disease risk and progression," *Scientific Reports*, vol. 8, no. 1, pp. 2112–12, 2018.

[15] L. Jena, S. Nayak, and R. Swain, "Chronic disease risk (cdr) prediction in biomedical data using machine learning approach," in *Advances in Intelligent Computing and Communication*, pp. 232–239, Springer, Cham, Switzerland, 2020.

[16] 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, 2016.

[17] J. Ramírez, J. Górriz, F. Segovia et al., "Computer aided diagnosis system for the alzheimer's disease based on partial least squares and random forest spect image classification," *Neuroscience Letters*, vol. 472, no. 2, pp. 99–103, 2010.

[18] A. Triastcyn and B. Faltings, "Bayesian differential privacy for machine learning," in *Proceedings of the International Conference on Machine Learning*, pp. 9583–9592, PMLR, Shenzhen, China, February 2020.

[19] S. Laur, H. Lipmaa, and T. Mielikäinen, "Cryptographically private support vector machines," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 618–624, Philadelphia, PA, USA, August 2006.

[20] K. Mandal and G. Gong, "Privfl: practical privacy-preserving federated regressions on high-dimensional data over mobile networks," in *Proceedings of the 2019 ACM SIGSAC Conference on Cloud Computing Security Workshop*, pp. 57–68, London, UK, November 2019.

[21] M. Shen, X. Tang, L. Zhu, X. Du, and M. Guizani, "Privacy-preserving support vector machine training over blockchain-based encrypted iot data in smart cities," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7702–7712, 2019.

[22] X. Liu, R. H. Deng, K.-K. R. Choo, and Y. Yang, "Privacy-preserving outsourced clinical decision support system in the cloud," *IEEE Transactions on Services Computing*, vol. 14, no. 1, pp. 222–234, 2017.

[23] Y. Zheng, H. Duan, C. Wang, R. Wang, and S. Nepal, "Securely and efficiently outsourcing decision tree inference," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 3, pp. 1841–1855, 2022.

[24] S. Tan, B. Knott, Y. Tian, and D. J. Wu, "Cryptgpu: fast privacy-preserving machine learning on the gpu," in *Proceedings of the 2021 IEEE Symposium on Security and Privacy (SP)*, pp. 1021–1038, IEEE, San Francisco, CA, USA, May 2021.

[25] F. Zheng, C. Chen, and X. Zheng, "Towards secure and practical machine learning via secret sharing and random permutation," 2021, https://arxiv.org/abs/2108.07463.

[26] X. Li, J. He, P. Vijayakumar, X. Zhang, and V. Chang, "A verifiable privacy-preserving machine learning prediction scheme for edge-enhanced hcpss," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 8, pp. 5494–5503, 2022.

[27] Z. Ma, J. Ma, Y. Miao et al., "Lightweight privacy-preserving medical diagnosis in edge computing," *IEEE Transactions on Services Computing*, vol. 15, no. 3, pp. 1606–1618, 2022.

[28] L. Sun, W.-S. Mu, B. Qi, and Z.-J. Zhou, "A new privacy-preserving proximal support vector machine for classification of vertically partitioned data," *International Journal of Machine Learning and Cybernetics*, vol. 6, no. 1, pp. 109–118, 2015.

[29] R. Chen, Q. Xiao, Y. Zhang, and J. Xu, "Differentially private high-dimensional data publication via sampling-based inference," in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 129–138, Sydney Australia, August 2015.

[30] J. Zhang, X. Xiao, Y. Yang, Z. Zhang, and M. Winslett, "Privgene: differentially private model fitting using genetic algorithms," in *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pp. 665–676, New York, NY, USA, June 2013.

[31] O. Ohrimenko, F. Schuster, C. Fournet et al., "Oblivious multi-party machine learning on trusted processors," in *Proceedings of the 25th {USENIX} Security Symposium ({USENIX} Security 16)*, pp. 619–636, Austin TX USA, August 2016.

[32] K. A. Jagadeesh, D. J. Wu, J. A. Birgmeier, D. Boneh, and G. Bejerano, "Deriving genomic diagnoses without revealing patient genomes," *Science*, vol. 357, no. 6352, pp. 692–695, 2017.

[33] H. Yu, J. Vaidya, and X. Jiang, "Privacy-preserving svm classification on vertically partitioned data," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 647–656, Springer, Cham, Switzerland, 2006.

[34] Z. Brakerski and V. Vaikuntanathan, "Fully homomorphic encryption from ring-lwe and security for key dependent messages," in *Annual Cryptology Conference*, pp. 505–524, Springer, 2011.

[35] R. Bost, R. A. Popa, S. Tu, and S. Goldwasser, "Machine learning classification over encrypted data," in *Proceedings of the Internet Society Network and Distributed System Security Symposium*, pp. 1–4, San Diego, CA, USA, February 2015.

[36] J.-C. Bajard, P. Martins, L. Sousa, and V. Zucca, "Improving the efficiency of svm classification with fhe," *IEEE*

Transactions on Information Forensics and Security, vol. 15, pp. 1709–1722, 2020.

[37] J. Wang, L. Wu, H. Wang, K.-K. R. Choo, and D. He, "An efficient and privacy-preserving outsourced support vector machine training for internet of medical things," *IEEE Internet of Things Journal*, vol. 8, no. 1, pp. 458–473, 2021.

[38] M. Zhang, W. Song, and J. Zhang, "A secure clinical diagnosis with privacy-preserving multiclass support vector machine in clouds," *IEEE Systems Journal*, vol. 16, no. 1, pp. 67–78, 2022.

[39] T. Li, Z. Huang, P. Li, Z. Liu, and C. Jia, "Outsourced privacy-preserving classification service over encrypted data," *Journal of Network and Computer Applications*, vol. 106, pp. 100–110, 2018.

[40] B. Xie, T. Xiang, X. Liao, and J. Wu, "Achieving privacy-preserving online diagnosis with outsourced svm in internet of medical things environment," *IEEE Transactions on Dependable and Secure Computing*, p. 1, 2021.

[41] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proceedings of the International conference on the theory and applications of cryptographic techniques*, pp. 223–238, Springer, Prague, Czech Republic, May 1999.