SEMMI: Multi-party Security Decision-making Scheme Under ~~the~~ Internet of Medical Things*

Cheng Li[a], Li Yang[a,*], Shui Yu[b], Wenjing Qin[a], Jianfeng Ma[c,d]

[a]*School of Computer Science and Technology, Xidian University*
[b]*School of Software, University of Technology Sydney*
[c]*School of Cyber Engineering, Xidian University*
[d]*Guizhou University*

Abstract

In ~~the~~ Internet of Medical Things, the intelligent auxiliary decision-making system uses machine learning algorithms to analyze medical data, which can effectively help patients and doctors analyze their health conditions. However, due to the ~~particularity~~ of the medical industry, its requirements ~~for~~ data security protection are ~~more~~ stringent, which leads to the reduction of data circulation, and related applications are difficult to be widely used. Therefore, this paper proposes an intelligent classification decision-making program in ~~the~~ Internet of Medical Things - SEMMI, which can effectively deal with the risk of data leakage in the process of medical data processing. At the same time, the huge computing and storage pressure caused by encryption and decryption operations in medical institutions is relieved.

In this scheme, data collection, network transmission, and calculation processes are all carried out under ciphertext. Since the resource allocation of each participant is different, we use chaos theory to construct a stream cipher algorithm on the sensor side to ensure the security of sensor-to-user transmission; in users, trusted organizations (large medical institutions) and the cloud, we use homomorphic encryption algorithm. In this way, the computability and storage security of the ciphertext can be guaranteed. Through security analysis

*This paper is an expanded version of the ICC2022.

*Corresponding author
Email address: yangli@xidian.edu.cn (Li Yang)

and experiments, the scheme can resist the attack of the adversary, and at the same time effectively reduce the calculation and network transmission pressure of each participant. Finally, we also discuss the flexibility of the scheme, and the results show that our scheme can be well applied in other algorithms as well.
Keywords: , Internet of Medical Things, Data Security, Intelligent Decision System, Homomorphic Encryption
2010 MSC: 00-01, 99-00

1. Introduction

With the development of sensors, Internet of Things, 5G networks, etc., applications such as Smart Homes, Smart City, and Wise Information Technology of med have also been widely promoted. At the same time, the rapid development of sensor computing, network transmission, and miniaturized equipment also has a certain role in promoting the Internet of Medical Things (IoMT). IoMT can process a large amount of data collected by sensors (such as blood pressure, heart rate, pulse), apply machine learning, neural network and other algorithms to mine the potential information contained in the data, and provide auxiliary reference for real-time decision-making[2]. It can bring people great convenience and precise services[3]. However, the computing power, storage capacity, etc. of sensor nodes deployed in wireless networks are insufficient. Therefore, cloud-based storage systems can effectively reduce the storage and computing overhead of users[4].

In applications such as smart medical and wearable devices, the data flow process can be roughly applied to the following three scenarios: (1) Use sensor devices to collect user biometric data, and then send these data to medical institutions or cloud storage. At the same time, these data will be used for remote diagnosis, such as: telemedicine, user health self-diagnosis, etc. (2) Community hospitals and private clinics need to send data to large medical institutions for authoritative auxiliary diagnosis due to factors such as personnel and equipment level. (3) Insurance and financial institutions need to know the health

status of their clients when conducting business, but these institutions do not have the ability to detect and diagnose, and need other medical institutions to help them make judgments[5, 6, 7]. Therefore, these institutions need to send customer information to the medical institution, and then the medical institution will give the test results based on the data. In medical institutions, there are many applications that use machine learning and neural network algorithms to process, analyze, and obtain results to assist medical personnel in medical diagnosis[8]. For example, Xing et al.[9] improved the KNN algorithm in the context of smart medical care to make up for the algorithm's deficiencies in processing large data sets. Elhoseny et al.[1] used Self Organizing Maps (SOM) and Optimal Recurrent Neural Networks (ORNN) to classify ovarian cancer in women. It can be seen from the experimental results that the program is more prepared, which will effectively improve the probability of disease detection.

However, with the indepth research on related technologies and applications[10, 11], there are many problems that threaten user privacy and security in the application process of IoMT. First, due to the limited resources of data acquisition equipment[12], it is easy to be attacked by adversaries. Second, recent studies have found that machine learning and neural network algorithms are very likely to cause data leakage during the training process, thereby posing a threat to user privacy. A large amount of data can help companies bring users a more convenient and faster experience; at the same time, the private information implicit in the data is also something users do not want others to obtain. Relevant laws and regulations have also made clear provisions on the use, management, storage and other aspects of data, such as: GDPR promulgated by the European Union[13],and many more. How to use data security and rational has become an urgent problem[14].

1.1. Related work

In response to the problems described above, the majority of scholars have carried out relevant research in order to find a balance between data use and privacy protection. In 2017, Mohassel et al.[15] proposed the first privacy pro-

tection computing protocol used in machine learning algorithms, sending data to two servers in a non-competitive relationship, and using secure two-party computing to jointly train various Machine Learning Model. Yang et al.[16] published a review paper on federated learning, which mentioned three parameter exchange modes, which can complete the machine learning training process without sending or uploading user data.

Hasan et al.[17] are difficult to apply encryption algorithms such as RSA for IoMT data acquisition equipment, so a lightweight encryption algorithm is proposed to protect patient privacy. Hamza et al.[18] used a chaotic system to build a fast and secure image encryption system, which can effectively protect the privacy of image data in IoMT.

Liu et al.[19] used two participants to construct triples and outsourced data encryption to two untrusted clouds. In the process of calculation and query using the KNN classification algorithm, the data and results are security. Wang et al.[20] designed a secure support vector machine outsourcing computing solution in the smart medical scheme, which proved to be security in an honest and curious environment. In order to reduce the computing cost of individuals and enterprises, Wu et al.[21] proposed a cloud database ciphertext calculation scheme, and conducted experiments based on the KNN algorithm[22]. Zhou et al.[23] used differential privacy and Paillier encryption algorithm to implement a federated learning scheme under fog computing, which can complete the neural network training process without multiple participants exchanging data. Yang et al.[24] designed a data outsourcing KNN classification algorithm based on vector homomorphism algorithm, and verified it on multiple data sets. In addition to the above-mentioned work, there have been many researches based on homomorphic encryption[25, 26], secure two-party (multiple) computing[27, 28, 29], etc. designed neural network, linear regression, SVM, KNN and other model training programs under ciphertext data. At the same time, due to the huge amount of data stored in the cloud. Dishonest cloud service providers also pose a huge threat to data security and privacy.

1.2. Problems and challenges

As shown in Fig. 1, medical institutions give patients early warning or treatment based on real-time patient data. The query user will submit the user data it owns to the medical institution for testing, and determine whether to provide the user with corresponding services (such as simple treatment measures, medical insurance, credit loans, etc.) based on the results of the medical institution's testing.

Throughout the IoMT application process, the processing of data may face the following three privacy and availability issues:

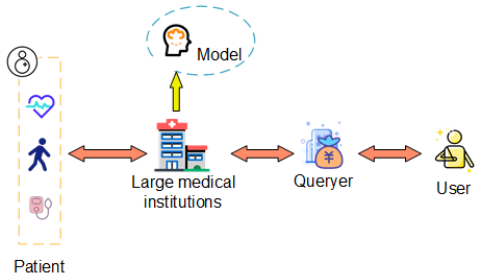

Figure 1: Framework diagram of the Internet of Medical Things

(1) The use of machine learning algorithms to classify results requires a large amount of data to participate, which is difficult for a single node or organization to complete.

(2) The data may be attacked by the adversary in the process of collection, transmission, processing, storage, etc. Including: untrusted cloud, malicious node, etc.At the same time, sensor nodes have limited resources, and it is necessary to find nodes with rich resources to provide them with computing, storage and other services[30].

(3) The data contains the user's private information, especially in some special application scenarios. For example: medical data, patients do not want their data to be known by others; at the same time, medical institutions cannot share the data with other institutions. If other institutions want to obtain information from their own data, they need to submit the data to the medical

institution for analysis. It can be done. In this scenario, medical institutions will be burdened in terms of computation, and it is also possible to leak the identity information of the user being detected.

1.3. Our contribution

In this paper, aiming at the security issues involved in the process of data , a security and efficient multi-party joint intelligent decision-making scheme is proposed in the IoMT - SEMMI (Security, Efficient, Medical IoT, Multi-party, Intelligent decision). First of all, in the scheme designed in this paper, all links are calculated and processed in the form of ciphertext; at the same time, taking into account the calculation pressure of all parties, the calculation process is outsourced to the cloud. Secondly, combined with the characteristics of data in practical applications, it is proposed to convert floating-point data into integer data for calculation, as far as possible to retain data information and improve classification accuracy. Specifically, the main contributions of this paper have the following three parts:

(1) In the scheme, each processing will be done by encrypted data. The scheme does not need to exchange the original data of users, and uses homomorphic encryption to ensure the security of data during processing and transmission. In order to ensure that the normal business of trusted institutions is not affected, the amount of calculation is reduced as much as possible.

(2) This paper analyzes the practicality of the algorithm and discusses deployment examples in other machine learning algorithms. At the same time, in view of the lack of deployment flexibility of the existing solutions, this paper also proposes a multi-key model, but it is currently limited by the computational overhead of related encryption algorithms.

(3) This paper evaluates the system's security, execution efficiency and accuracy of final results through theoretical and experimental methods. Since the encryption scheme can only operate on integers, in order to ensure the accuracy of the results, a method is proposed to increase the floating point number and then take the integer part to reduce the influence of this factor on the training

results. During the calculation and storage process, the Cloud does not know the meaning of the calculation results and the labels corresponding to the data itself, so the Cloud does not know anything about the information contained in the data.

1.4. Organization

The following will introduce the summary arrangements of the paper. In the second and third sections, some basic knowledge, system framework, threat model and design goals used in the thesis will be introduced. In the fourth section, we will discuss the main algorithms of the system. In the fifth section, the security and performance of the system will be analyzed. Section VI will summarize the research work of this paper.

2. Preliminary

First, introduce the symbols in this paper. $\mathbb{Z}$ is defined as the set of integers, and $\mathbb{N}$ is defined as the set of natural numbers. $a \xleftarrow{R} \chi$ means that a is selected randomly under the χ distribution, and $b \xleftarrow{U} \mathbb{Z}$ means that the only b is selected in the set of integers $\mathbb{Z}$. Use uppercase and bold letters to represent the matrix, for example: $\mathbf{M}$; use lowercase and bold letters to represent the vector, for example: $\mathbf{v}$, v_i represents the i-th element in the vector. It is specifically stated here that $\mathbf{G}$ and $\mathbf{G}^{-1}$ are not matrices and their inverse matrices, but specific operations, such as: $\mathbf{G} = \mathbf{I} \cdot \mathbf{g} = \begin{bmatrix} 1 & 2 & 4 & & & & & 0 & \\ & & & 1 & 2 & 4 & & & \\ & 0 & & & & & 1 & 2 & 4 \end{bmatrix}$, where $\mathbf{g} = 2^0, 2^1, 2^2$, so there is $\mathbf{G}\mathbf{G}^{-1}\mathbf{M} = \mathbf{M}$

2.1. Homomorphic encryption and GSW scheme

1) Homomorphic encryption is a form of encryption, which allows the ciphertext to be subjected to a specific form of algebraic operation, and the result obtained is the same as the result calculated in the plaintext[31]. The homomorphic encryption scheme mainly includes four algorithms (*KeyGen*, *Enc*, *Dec*, *Eval*), which will be briefly introduced in the following.

$KeyGen(1^\lambda) \rightarrow (sk, pk, evk)$: Given the initialization parameters, the algorithm will output the private key sk, the public key pk,, and the public transformation key evk required by the homomorphic encryption system.

$Enc(m, pk) \rightarrow (c)$: Encrypt the plaintext information m with the given public key pk to obtain the ciphertext c.

$Dec(c, sk) \rightarrow (m)$: Decrypt with the given private key sk and ciphertext m, and finally get the plaintext m.

$Eval(f, c_1, c_2, \cdots, c_x) \rightarrow (c_{eval})$: Given algebraic operations f and $c_1, c_2, \cdots, c_x$ ciphertexts, get the ciphertext calculation result c_{eval}.

(2) GSW homomorphic encryption scheme [32]. The homomorphic encryption scheme was proposed by Gentry, Sahai, and Waters in 2013. Since the encryption scheme proposed by Gentry et al. can only encrypt a single bit of data, its encryption efficiency is low. After that, [33, 34, 35, 36] proposed improvements to it, enabling it to perform encryption operations on integer matrices. Next, the GSW program will be briefly introduced.

$GSW.Setup(1^\lambda, 1^d) \rightarrow (pp)$: First, select the grid dimension parameter as $n = n(\lambda, d)$, the noise distribution as $\chi = \chi(\lambda, d)$, the integer $q = q(\lambda) \geq 2$, and the random matrix $\mathbf{B} \in \mathbb{Z}_q^{(n-1)\cdot m}$, and the output parameter $pp = \{n, \chi, q, \mathbf{B}, \mathbf{G}\}$.

$GSW.KeyGen(pp) \rightarrow (sk, pk)$: In the GSW encryption scheme, first select $\mathbf{s} \xleftarrow{R} \mathbb{Z}_q^{n-1}$, and then generate the private key $sk = \mathbf{t} = (-\mathbf{s}, 1) \in \mathbf{Z}_q^n$; then the public key generation process, randomly select random noise $\mathbf{e} \leftarrow \chi^m$, and then calculate $\mathbf{b} = \mathbf{sB} + \mathbf{e} \in \mathbb{Z}_q^m$ to obtain the public key $pk = \mathbf{A} = \begin{bmatrix} \mathbf{B} \\ \mathbf{b} \end{bmatrix} \in \mathbb{Z}_q^m$.

$GSW.Enc(msg, pk) \rightarrow (\mathbf{c})$: First, randomly select the matrix $\mathbf{R} \xleftarrow{R} \{0,1\}^{m \times m}$ and the message $msg \in \{0, 1\}$, and then calculate the ciphertext matrix $\mathbf{C} = \mathbf{AR} + (msg)\mathbf{G} \in \mathbb{Z}_q^{n \times m}$.

$GSW.Dec(\mathbf{C}, sk) \rightarrow (msg')$: First, define the vector $\mathbf{v} = [0, 0, \ldots, 0, \lceil q/2 \rceil]$, then calculate $v = \mathbf{tGG^{-1}}(\mathbf{w^T}) \in \mathbb{Z}_q^m$, and finally output $msg' = | \lceil v/(q/2) \rfloor |$.

$GSW.Add(\mathbf{C_1}, \mathbf{C_2}) \rightarrow (\mathbf{C_3})$: Given two ciphertext matrices $\mathbf{C_1}$ and $\mathbf{C_2}$, output their sum $\mathbf{C_3} = \mathbf{C_1} + \mathbf{C_2} \in \mathbb{Z}_q^{n \times m}$.

$GSW.Mult(\mathbf{C_1}, \mathbf{C_2}) \rightarrow (\mathbf{C_3})$: Given two ciphertext matrices $\mathbf{C_1}$ and $\mathbf{C_2}$, output their product $\mathbf{C_3} = \mathbf{C_1}\mathbf{G^{-1}}\mathbf{C_2} \in \mathbb{Z}_q^{n \times m}$.

3. Scheme model and Design Goal

In this section, we will first introduce the SEMMI system model, including the participants and their functions in the system. The threat model mainly introduces the security threats that the system faces or can resist. Finally, four important goals that can be achieved by the scheme proposed in this paper will be introduced.

3.1. Scheme model

In the program, there are five types of participants. *Trusted organization.* They can also be called service providers, such as hospitals, physical examination centers and other institutions, which have a large amount of sample information and user data and can provide services to other users. *Sensor equipment.* It is installed by the user or a medical institution, and the sensor collects data for the user and monitors the user's physical health. *Users.* Collect their own information through sensor equipment, and at the same time send useful data after pre-processing the collected information to a trusted organization, and the trusted organization will complete the diagnosis of the user. *Query organization.* This type of users may come from community hospitals, private clinics, insurance, finance and other fields. By submitting the collected data and sending them to trusted institutions for testing, they can obtain corresponding health information. *Cloud.* The participant is untrustworthy. Although it has large processing, computing, and storage capabilities, it may also spy on the data stored by the user.

Next, the program execution process will be introduced. As shown in Fig. 2:

(1) In the system initialization phase. A trusted institution (medical institution) generates the parameters, public key, and private key data in the process of system operation, encryption, and decryption. The flow cipher key between the sensor and the user is set by the user.

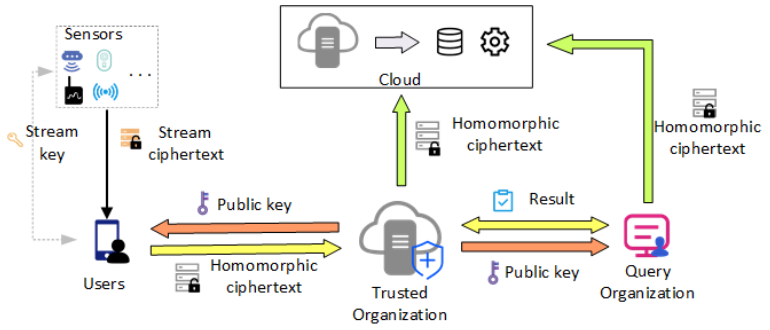

Figure 2: SEMMI system framework

The sensor equipment is located at the bottom of the scheme and is used to collect user-related physical data. In order to reduce the calculation and communication overhead of the sensor, the sensor encrypts the collected data in the form of a one-dimensional vector $\overrightarrow{v}$ to generate a ciphertext $C_s(\overrightarrow{v}_i)$, and then transmits it to the user for decryption and processing. Finally, the encrypted data $C(\overrightarrow{v}_i)$ is stored in Cloud computing server. Since the encryption scheme is only applicable to integer data types, in order to reduce the impact of the loss of the decimal part of the data on the accuracy of the final result, part of the floating-point data is amplified.

(2) Since trusted institutions and query institutions cannot share their own medical data, they cannot bear the large amount of computational expenses that they bring. Therefore, the data needs to be encrypted and then calculated by a third party, and there is no direct data interaction between the two parties. In this way, data privacy can be guaranteed not to be leaked, and data from both parties can be used to make predictions. When the query user applies for service, the trusted agency distributes the encrypted data public key pk. After that, the inquiring user encrypts the data he owns and uploads it to the cloud. Then, the data is calculated by the cloud, the decryption of the data is pre-decrypted by the trusted organization, and finally the result of the pre-decryption is sent to the query organization. The query user calculates the pre-decrypted data to obtain the final data result.

This program consists of four phases, namely: (1) Use stream encryption for data transmission between the sensor and the user.

(2) The user obtains public key information from a trusted institution, encrypts the data and uploads it to the cloud.

(3) The query user requests services from the trusted agency, the trusted agency sends the public key to the query user, the query user uses the public key to encrypt the data and then sends the data to the cloud for calculation, and the cloud feeds back the calculation result to the trusted agency.

(4) The trusted organization decrypts the calculation result and sends it to the query organization, and the query organization performs statistics on the result to obtain the final decision result.

3.2. Threat Model

As shown in Fig. 2, there are a total of five participants in the scheme, among which the trusted institution is completely credible, and the other four participants can be untrusted. Therefore, we assume an adversary $\mathcal{A}$, which may obtain data information from users other than trusted institutions. We assume the following possible situations:

(1) The adversary $\mathcal{A}$ can obtain the encrypted data of all parties by monitoring the network communication link.

(2) The adversary $\mathcal{A}$ can launch an attack on the cloud to obtain the encrypted data stored by the user in the cloud.

(3) The adversary $\mathcal{A}$ can unite some of its participants to infer other users' plaintext information through ciphertext.

3.3. Project objectives

In this scheme, we will achieve four goals to address security and efficiency issues in specific scenarios.

Data security: Since medical data contains a large amount of patient information, it is easy to obtain the correspondence between patient identity and

data through reasoning. Therefore, the designed scheme should reduce the original data exchange between the participating parties as much as possible, and it is best to complete the training prediction and task without exchanging the original data.

Distributed system requirements: each participant in the system may be in different regions, countries, etc. Therefore, it is necessary to build a distributed training and prediction system to meet the service needs of users in different regions. At the same time, the distributed environment can also obtain more data, thereby improving the accuracy of prediction classification.

Efficient operation: In the operation of the system, due to the limited computing, storage, and processing capabilities of each participant. Therefore, it is necessary to outsource the computing tasks of each participant as far as possible to other computing nodes with stronger computing capabilities or more free resources, so that other computing tasks of each participant are not affected.

Accurate classification: In the application process of the scheme, the results of predictive classification cannot be prevented from affecting the normal use of the system, that is, there can be no major differences compared to the results of classification using the original plaintext information.

4. The proposed scheme

In this section, the specific implementation details of the SEMMI program will be introduced. The execution relationship of each stage is shown in Fig. 3:

4.1. User's data collection

Phase 1: The sensor uses stream encryption to transmit data to the user.

First, generate the corresponding $key := X_n$ according to the formula $X_{n+1} = \mu \cdot X_n \cdot (1 - X_n)$. During the experiment, the parameters adopt $\mu = 3.6$, the initial value $X_1 = 0.6316$, and the number of iterations select $n = 256$. Then, input the *key* and the transmitted plaintext information $\vec{M}$ into the encryption algorithm to generate the corresponding ciphertext $\vec{C}$. Finally, it is sent to the

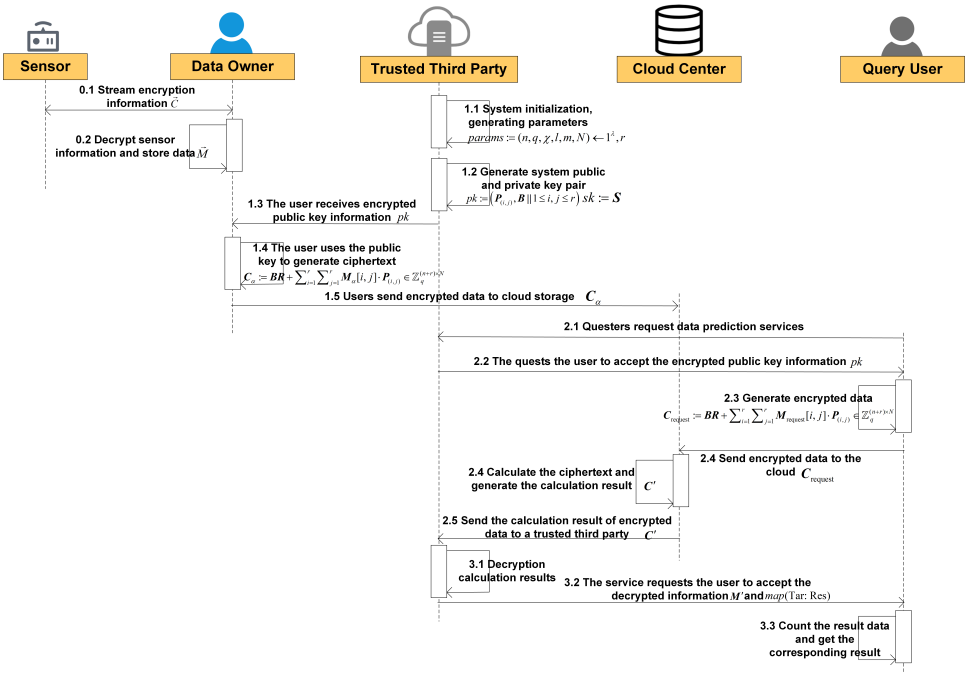

Figure 3: SEMMI system framework

user, and the user uses the same parameters to generate a key, decrypts the ciphertext $\overrightarrow{C}$, and obtains the corresponding plaintext $\overrightarrow{M}$.

4.2. User encrypted data upload and query service request

There are two phases in this section: (1) User encrypted data upload; (2) The query user issues a service request and uploads the data, and then the cloud calculates the ciphertext data. Among them, the trusted organization establishes the mapping relationship *map(Tar:(Id-Data))* between the data and the result tag, and then uploads the encrypted data and the key value *map(Id:C(Data))* to the cloud for storage. After the cloud computing is completed, the key-value pair *map(Id:Res)* composed of the key value and the result ciphertext is returned to the trusted agency, and the trusted agency decrypts the ciphertext and returns the key-value pair *map(Tar:Res)* composed of the result and the result tag to the test user. Test user statistics and get the corresponding *result*.

Initialization:

Setup($params \leftarrow 1^{\lambda}, r$): First, determine the security parameter λ and the depth L of the multiplication circuit; then, according to the security parameter λ, determine the dimension n of the integer lattice, an integer modulus q, and an χ in $\mathbb{Z}$ to obey the sub-Gaussian distribution. Let $l := \lceil log_2 q \rceil$, $m := O((n+r)l)$, $N := (n+r) \cdot l \cdot r$ is the length of the plaintext vector. The ciphertext space is $\mathbb{Z}_q^{(n+r)\times N}$. There are $\mathbf{g}^T = (1, 2, \ldots, 2^{l-1})$, $\mathbf{G} = \mathbf{g}^T \otimes \mathbf{I}_{n+r}$.

$$params := (n, q, \chi, l, m, N) \leftarrow 1^{\lambda}, r \tag{1}$$

Key generation:

$KeyGen(1^{\lambda}, r)$: Randomly select the unique matrix $\mathbf{A} \xleftarrow{U} \mathbb{Z}_q^{n\times m}$ the private key matrix $\mathbf{S}' \xleftarrow{R} \chi^{r\times n}$, and the noise matrix $\mathbf{E} \xleftarrow{R} \chi^{r\times m}$. Set the size of the identity matrix $\mathbf{I}_r$ as $r \times r$. $\mathbf{B} := ((\mathbf{S}'\mathbf{A} + \mathbf{E})/\mathbf{A}) \in \mathbb{Z}_q^{(n+r)\times m}$. $\mathbf{M}_{(i,j)}$ is the matrix of $r \times r$, its (i, j) position is 1, and the other positions are 0.

Therefore, we get:

$$\mathbf{S} = [\mathbf{I}_r \parallel -\mathbf{S}'] \in \mathbb{Z}_q^{r\times(n+r)} \tag{2}$$

$$\mathbf{P}'_{(i,j)} = [\frac{\mathbf{M}_{(i,j)S}}{\mathbf{0}}]\mathbf{G} \in \mathbb{Z}_q^{r\times(n+r)} \tag{3}$$

$$\mathbf{P}_{(i,j)} = \mathbf{BR}_{(i,j)} + \mathbf{P}'_{(i,j)} \tag{4}$$

$$pk := (\mathbf{P}_{(i,j)}, \mathbf{B} \parallel 1 \le i,j \le r), sk := \mathbf{S} \tag{5}$$

Phase 2: The trusted organization sends the public key to all users of the data. The user encrypts his data with the public key and stores the encrypted data in the cloud.

$PubEnc(\mathbf{C} \leftarrow pk, \mathbf{M}_\alpha)$: The plaintext matrix of user α is $\mathbf{M}_\alpha$, and the matrix $\mathbf{R} \xleftarrow{U} (0,1)^{m\times N}$ is randomly selected to calculate the ciphertext

$$\mathbf{C}_\alpha := \mathbf{BR} + \sum_{i=1}^{r}\sum_{j=1}^{r}\mathbf{M}[i,j]\cdot\mathbf{P}_{(i,j)} \in \mathbb{Z}_q^{(n+r)\times N} \tag{6}$$

The user uploads $\mathbf{C}_\alpha$ to the cloud for storage and calculation.

Phase 3: The query user requests services from the trusted agency, encrypts the data by using the public key, and then uploads the ciphertext to the cloud for calculation, and the cloud sends theciphertext of the calculation result to the trusted agency.

$PubEnc(\mathbf{C} \leftarrow pk, \mathbf{M}_{request})$: The plaintext matrix information of the query organization is $\mathbf{M}_{request}$, and the matrix $\mathbf{R} \xleftarrow{U} (0,1)^{m\times N}$ is randomly selected to calculate the ciphertext

$$\mathbf{C}_{request} := \mathbf{BR} + \sum_{i=1}^{r}\sum_{j=1}^{r}\mathbf{M}_{request}[i,j]\cdot\mathbf{P}_{(i,j)} \in \mathbb{Z}_q^{(n+r)\times N} \tag{7}$$

Then, the ciphertext $\mathbf{C}_{request}$ calculated by the query user is uploaded to the cloud for calculation.

$Add(\mathbf{C}'_{add} \leftarrow \mathbf{C}_\alpha, (-\mathbf{C}_{request}))$: After the cloud receives the ciphertext calculation request from the query user, add $\mathbf{C}_\alpha$ and $\mathbf{C}_{request}$ to get $\mathbf{C}'_{add}$.

$Mult(\mathbf{C}' \leftarrow \mathbf{C}'_{add}, \mathbf{C}'_{add})$: After the cloud receives the ciphertext calculation request from the query user, multiply $\mathbf{C}'_{add}$ to get the distance of each dimension feature of the data, $\mathbf{C}' = \mathbf{C}'_{add}\mathbf{G}^{-1}, \mathbf{C}'_{add} \in \mathbb{Z}_q^{(n+r)\times N}$

The cloud then sends $\mathbf{C}'$ to the trusted authority for pre-decryption calculation.

4.3. Decryption of data

Phase 4: The trusted organization pre-decrypts the result data, and then sends the result (intermediate parameter) to the query user, and the query organization performs statistics and calculates the final result.

$Dec(\mathbf{M} \leftarrow \mathbf{C}, \mathbf{S})$: First: define $\mathbf{O} = \mathbf{SC} = \mathbf{MSG} + \mathbf{E} \in \mathbb{Z}_q^{r \times N}$.

After that, pack the first rl column into a matrix $\mathbf{O}'$, and the noise contained in it is $\mathbf{E}'$.

Then, the decryption operation will be performed. For $\mathbf{M}[i, j] = \sum_{k=0}^{l-2} 2^k \cdot x_k$, where x_k is the value of the k bit after x is binary-encoded, the pre-decrypted data $\mathbf{M}'$ is finally generated, and $\mathbf{M}'$ and $map(tar : Res)$ are sent to the query user.

The query user can successfully obtain the distance of each dimension feature of the plaintext data through the pre-decryption results $\mathbf{M}'$ and $map(tar : Res)$ of the trusted organization. After that, the final prediction result can be obtained by query user through the distance and the data label given by the trusted organization.

4.4. The idea of multi-key construction to increase the flexibility of the scheme

For devices with richer computing resources, we have also improved the above algorithm and designed a more flexible encryption and decryption algorithm.

As shown in Fig. 4, we design a security model training scheme under multi-key homomorphic encryption. In the multi-key mode, the functions of each participant are basically the same as the previous scheme. But the computational task of trusted organization is lightened. Here, the trusted organization only needs to assign their respective public and private keys to the participants. The query user can also independently decrypt the training results.

Next, we will introduce the training process under multi-key. Since the data collection part is the same as the previous algorithm, it will not be repeated

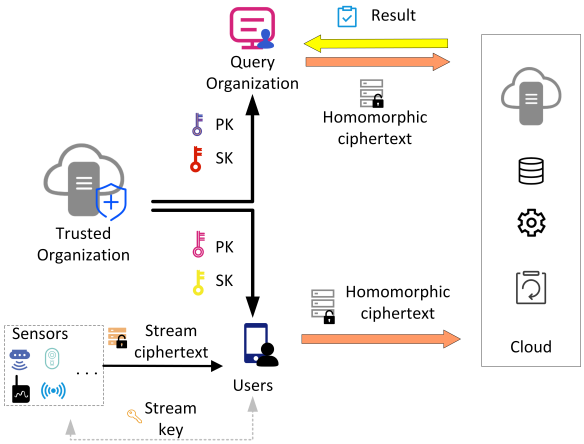

Figure 4: System model under multi-keys.

here. In the IoMT, the number of participants is difficult to predict. Therefore, keys of a certain size cannot be generated in advance[37]. This flexibility, system energy consumption, scalability will be restricted. In order to ensure that each participant can join at any time, we refer to [38, 39] to construct a multi-key scheme.

$TrustedOrganizationSetup$: First, a unique matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ is randomly generated by a trusted organization.

$TrustedOrganizationInit - KeyGen(\mathbf{A})$: A trusted organization generates an initialization key $\mathbf{t}$ for one of the users. $\mathbf{t} = (-\bar{\mathbf{t}}, 1) \in \mathbb{Z}^n$, where $\bar{\mathbf{t}} \leftarrow \chi^{n-1}$. $\mathbf{e} \leftarrow \chi^m$, from which can get $\mathbf{b} := \mathbf{tA} + \mathbf{e} \approx \mathbf{tA} \in \mathbb{Z}_q^m$. Therefore, $\mathbf{t}$ is the private key and $\mathbf{b}$ is the public key used for extending ciphertext.

$UserEnc(\mathbf{t}, \mu \in \{0,1\})$: First, find an LWE matrix $\bar{\mathbf{C}} \in \mathbb{Z}_q^{n \times nl}$ such that $\mathbf{t}\bar{\mathbf{C}} \approx 0$ holds. Therefore, $\mathbf{C} := \bar{\mathbf{C}} + \mu(\mathbf{I}_n \otimes \mathbf{g}) \in \mathbb{Z}_q^{n \times nl}$. Then, the matrix $\mathbf{R} \in \{0,1\}^{m \times nl}$ is randomly selected to define $\mathbf{F} := \mathbf{AR} + \mu(\mathbf{I}_n \otimes \mathbf{g}) \in \mathbb{Z}_q^{n \times nl}$. Finally, the matrix $\bar{\mathbf{D}} \in \mathbb{Z}_q^{nml \times nl}$ is chosen, where $(\mathbf{I}_{ml} \otimes \mathbf{t}) \cdot \bar{\mathbf{D}} \approx \mathbf{0}$. Therefore, $\mathbf{D} := \bar{\mathbf{D}} + (\mathbf{R} \otimes \mathbf{g}^T \otimes \mathbf{e}_n^T)$ can be obtained. Therefore, the ciphertext group will be obtained ($\mathbf{C}$, $\mathbf{F}$, $\mathbf{D}$)

$UserCTExt(\mathbf{t}_i^*, \mathbf{C}, \mathbf{F}, \mathbf{D})$: There are mainly three steps here. The first step is to convert the key $\mathbf{t}_{i-1}$ to the key $\mathbf{t}_i$ containing the current user, the sec-

ond step is to convert the ciphertext $\mathbf{C}^*_{i-1}$ sent by the previous user into the ciphertext $\mathbf{C}_{i-1}$ corresponding to $\mathbf{t}_i$, and the third step is to convert the current user. The ciphertext data $\mathbf{C}^*_i$ corresponding to the held plaintext is added (multiplied) to $\mathbf{C}_{i-1}$.

$UserCTExt-1$: Suppose the private key of user i is $\mathbf{t}^*_i$, therefore, $\mathbf{b}^*_i \approx \mathbf{t}^*_i\mathbf{A} \in \mathbb{Z}^m_q$ can be obtained. $\mathbf{t}^*_i$ converted key $\mathbf{t}_i = (\mathbf{t}_{i-1}, \mathbf{t}^*_i) \in \mathbb{Z}^{n'}$. $n' = ni$.

$UserCTExt-2$: Definition, $\mathbf{F'} = \mathbf{F}$, $\mathbf{R'} = \mathbf{R}$, $\mathbf{D'} := (\mathbf{I}_{ml} \otimes \begin{pmatrix} \mathbf{I}_{n'} \\ \mathbf{0}_{n\times n'} \end{pmatrix}) \cdot \mathbf{D} \in \mathbb{Z}^{(n+n')ml\times nl}_q$. Then, $\mathbf{C'} := \begin{pmatrix} \mathbf{C} & \mathbf{X} \\ \mathbf{0} & \mathbf{F} \end{pmatrix} \in \mathbb{Z}^{(n+n')ml\times nl}_q$. Among them, $\mathbf{s} := \mathbf{G}^{-1}(-\mathbf{b}^*) \in \{0,1\}^{ml}$, $\mathbf{X} := (\mathbf{s} \otimes \mathbf{I}_{n'}) \cdot \mathbf{D} \in \mathbb{Z}^{n'\times nl}$.

$UserCTExt-3$: Suppose there are two sets of ciphertexts: $(\mathbf{C}_1, \mathbf{F}_1, \mathbf{D}_1$, $(\mathbf{C}_2, \mathbf{F}_2, \mathbf{D}_2$. The final result is $(\mathbf{C'}, \mathbf{F'}, \mathbf{D'}$ obtained by adding (multiplying) the two sets of ciphertexts.

It is not difficult to see that the above algorithm increases the flexibility of solution deployment, but it will bring some overhead to users. The above scheme can construct a centerless system, that is, only the corresponding ciphertext needs to be stored in the cloud. But whether it is [37] or [38, 39] , there are certain limitations in designing a multi-key system. A large amount of computing still needs to be deployed on the trusted organization or user side, which will consume energy and other aspects for these participants.

Therefore, the ideas under this model are beneficial to the wide application of IoMT. However, due to the complex calculation of the relevant encryption algorithm and the low practicability at present, the deployment of this idea is limited.

5. Analysis and evaluation

In this section, we will evaluate the security and performance of the SEMMI system. First, the security of the solution is proved by the assumption of the security scenario. Then, the program is analyzed experimentally and tested on

real data sets and simulated data respectively to obtain the accuracy, calculation, and storage overhead of the system. Finally, the four security goals reached in this paper are summarized.

5.1. Security Analysis

First, we will first analyze the security of the encryption and decryption algorithms in this paper. Then, suppose that the adversary $\mathcal{A}$ can obtain the ciphertext and public key in the system, and even collude or control some nodes to launch an attack on the system.

1) Security analysis of encryption and decryption schemes

Theorem 1: The chaotic system and stream cipher provide security from the sensor device to the user.

Proof: This article uses the RC4 stream cipher. Although this stream cipher system has been proven to have security risks, there is no way to crack it when the key length is more than 128 bits. At the same time, although encryption based on chaos theory cannot prove its security well at present, it is still difficult to crack chaotic systems[40].

Theorem 2: In the scheme, the security of the FHE scheme First of all, our security assumptions are based on the difficult assumptions of LWE. We will certify that the encryption system is IND-CPA secure.

Proof: According to the scheme proposed in this paper, the challenger first generates the public key $pk = \left\{\mathbf{P}_{(i,j)}, \mathbf{B} \parallel 1 \leq i \leq j \leq r\right\}$ and private key $sk = \mathbf{S} = [\mathbf{I} \parallel -\mathbf{S}']$ required for encryption and decryption. Then, the adversary $\mathcal{A}$ selects two plaintext information m_0 and m_1, and sends the plaintext information to the challenger for encryption. When the challenger receives the two plaintext messages, he randomly selects $b = \{0, 1\}$, and then encrypts the plaintext message m_b,and the corresponding ciphertext message is $\mathbf{C}_b$. After that, the generated ciphertext $\mathbf{C}_b$ is sent to the adversary $\mathcal{A}$, and the adversary $\mathcal{A}$ guesses the plaintext information corresponding to the ciphertext $\mathbf{C}_b$ in polynomial time.

Because $\mathbf{A} \xleftarrow{U} \mathbb{Z}_q^{n\times m}$, $\mathbf{S}' \xleftarrow{R} \chi^{r\times n}$,and $\mathbf{E} \xleftarrow{R} \chi^{r\times m}$ are randomly selected, $\mathbf{B} := ((\mathbf{S}'\mathbf{A}+\mathbf{E})/\mathbf{A}) \in \mathbb{Z}_q^{(n+r)\times m}$is also random. $\mathbf{B}' \xleftarrow{R} \mathbb{Z}_q^{(r+n)\times m}$ is randomly selected, and the adversary $\mathcal{A}$ cannot distinguish between $\mathbf{B}$ and $\mathbf{B}'$. Furthermore, it can be concluded that the public key information generated by $\mathbf{B}$ and $\mathbf{B}'$ is also indistinguishable from the adversary $\mathcal{A}$. Next, based on the security proofs in [32] and [34], we can conclude that the challenger encrypts the plaintext message $\mathbf{m}_b$ to obtain the ciphertext $\mathbf{C}_b$, and further, constructs a plaintext matrix $\mathbf{M}_b$ from multiple $\mathbf{m}_i$, and constructs the noise $\mathbf{e}$ as The form of the matrix is $\mathbf{E}$, and the ciphertext matrix $\mathbf{C}_b := \mathbf{BR} + \sum_{i=1}^{r}\sum_{j=1}^{r}\mathbf{M}[i,j]\cdot\mathbf{P}_{(i,j)} \in \mathbb{Z}_q^{(n+r)\times N}$ is obtained by encryption. Assuming that $\mathbf{C}' \xleftarrow{U} \mathbb{Z}_q^{(n+r)\times N}$ is randomly selected, it is not difficult to conclude that $\mathbf{C}'$ and $\mathbf{C}_b$ are also indistinguishable, which belongs to the LWE problem. Therefore, it can be concluded that the scheme is IND-CPA security.

2) Security analysis of the system

Theorem3: The hypothesis of the scheme is that the cloud is honest but curious, and the confidentiality of data and user privacy will not be obtained by adversaries.

Proof: First of all, except for the cloud, no other nodes in the system can contact the data information of other nodes. Therefore, only the cloud will guess and infer user data out of curiosity. According to Theorem1, it can be concluded that if the cloud can crack the data of other nodes in polynomial time with only the ciphertext and public key, it can also crack any LWE-based encryption system in polynomial time.

Theorem4: In the proposed KNN classification algorithm, assuming that the adversary $\mathcal{A}$ controls some user nodes, the system can still guarantee the confidentiality of other users' data and will not be paralyzed.

Proof: Suppose that there is an adversary $\mathcal{A}$, which can be a node in the system or other nodes, which can control some user nodes in the system to launch attacks. Since the tag data is stored in a trusted organization, the adversary $\mathcal{A}$ cannot obtain the valid tag information corresponding to the data. It can be

concluded from Theorem2 that when some nodes in the system are controlled by the adversary $\mathcal{A}$, the confidentiality of the data of other user nodes in the system will not be affected. The cloud is honest and curious, and it will not reject the user's computing request, so the query request of the inquiring user will not be affected. Therefore, the system can be used normally when some nodes in the system can be controlled by the adversary $\mathcal{A}$.

3) Achievement of program goals

Data security: In the previous section, we have discussed and proved the encryption system and adversary's attack methods. In the security hypothesis and threat model proposed in this paper, the scheme is secure, so the security of user data in the system is guaranteed.

Distributed system requirements: In the solution proposed in this article, users do not need to stay online at all times. After the user encrypts the processed data, it is sent to a trusted organization for diagnosis, and then the trusted organization forwards the data. After that, users and trusted institutions are not required to perform any calculation and storage tasks, effectively reducing their system overhead. For the inquiring user, there is a prediction task to establish a connection with a trusted organization. It can be seen that in the entire program application process, data collection, storage, query and other links meet the needs of distributed systems.

Efficient operation and accurate classification: This goal will be demonstrated through experiments in 5.2.

5.2. Experiment and evaluation

Experimental environment: hardware device is Intel Xeon(R) Gold 5218R 2.10GHz 2core, 32GB memory; software environment is Windows 10, using Anaconda3, Python version is 3.8, programming tool is Pycharm 2021 , The program uses the numpy toolkit to complete the calculation process of the matrix in the program. The source code of this paper is referenced to [41] during the writing process, and the data sets in [42] and [43] are used for simulation experiments. The value of the security parameter q is $2^{31} - 1$. All data sets use 25% as the

test set and 75% as the training set.

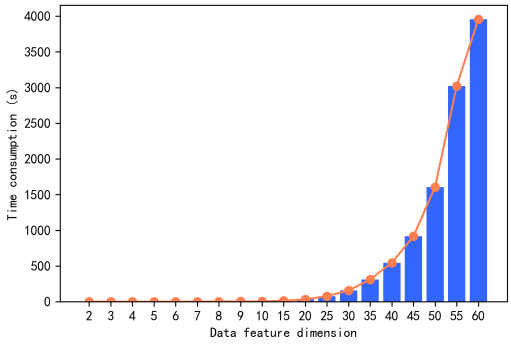

Figure 5: Key generation time consumption

We compared the time consumption of each stage of the encryption scheme. As shown in Fig. 5 and Fig. 6. As the dimension of the data features that need to be encrypted increases, the time overhead required for the system to generate the key will also increase. However, the computational time overhead required in the encryption and decryption process is relatively low. In the operation of the system, they will not cause excessive computing costs due to factors such as encryption and decryption.

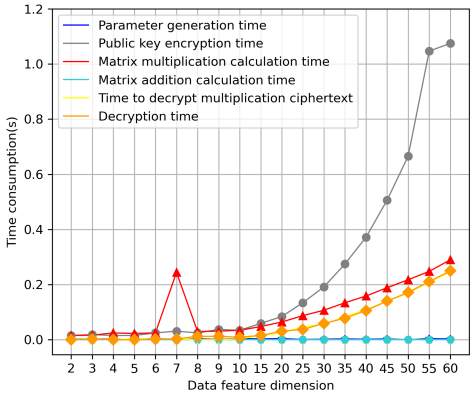

Figure 6: Encryption and decryption time consumption

As shown in Fig. 7, comparing with the encryption method of [23], it is not difficult to find that although the amount of ciphertext data in this scheme is relatively large, it has obvious advantages in the time consumption of encrypted data. With the increase of data dimensions, the increase in ciphertext in this paper is much smaller than that of Paillier's algorithm.

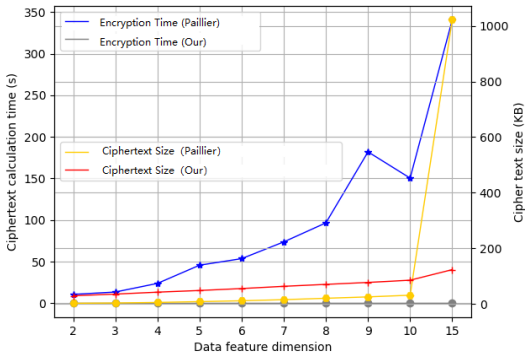

Figure 7: Comparison chart of ciphertext calculation time and required storage space

As can be seen in Fig. 8, the growth of the key storage space is directly proportional to the parameter setting. Therefore, under the premise of ensuring security, it is possible to select smaller parameters for calculation as much as possible, thereby reducing communication and storage overhead.

It is worth noting that because the data in some data sets has data between 0-1 or the accuracy of data in each dimension is quite different. For example: In the data set Fetal_health, some dimension values can reach 500, and some dimension data is 0.003. Processing such data may exceed the scope of digital processing. Therefore, in order to ensure that the data characteristics are not affected, based on matrix theory and geometric characteristics, this paper will shift and scale the data dimensions that are too small and too large. In this way, the relative distance between the sample points and test points in the data set will not be changed, thereby ensuring that the accuracy will not affect the accuracy of the test results due to the preprocessing of the data.

This paper conducted experiments on five data sets, as shown in table 1.

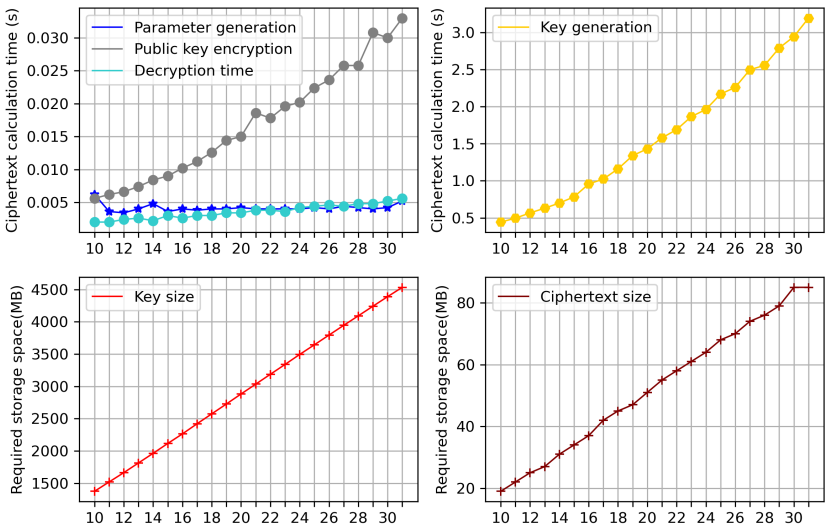

Figure 8: The relationship between the parameter q setting and the growth of key storage

Table 1: Accuracy of the system in real data sets

Data Set	KNN	KNN-1	KNN-10	KNN-100
Iris	1.00	1.00	1.00	1.00
Breast	0.925	0.937	0.970	0.9175
Diabetes	0.736	0.718	0.738	0.732
Column_2c_weka	0.855	0.836	0.8512	0.833
Fetal_health	0.896	0.880	0.892	0.896

In the experiment process, the data range is first transformed so that the data of all dimensions are in the same numerical range, and then the KNN classification prediction algorithm is performed. In the Iris data set, due to its small amount of data and fewer feature dimensions, the prediction results are better. Experiments in other datasets will result in some loss of data in KNN-1, so the accuracy will decrease; while in KNN-10, the accuracy of the data will increase, so the classification accuracy will also increase. But for KNN-100, the test accuracy of some data sets shows a downward trend. This is because after some values are increased by a factor of 100, the calculation result of the square of the value may be too large when running in the program, which leads to loss of accuracy when calculating the distance between two points. If the data overflow during the experiment is not considered, the accuracy of the result will increase as the data increases, and it will eventually approach the accuracy of plaintext training.

Next, use the UCI breast cancer data set to compare the computational time costs of each participant in the program. There are 31-dimensional features in the data set, and there are two classification result labels: "malignant" and "benign".

Table 2: The time consumption of each participant in the Breast dataset

Party	Time(s)
Data provider users	87.9097
Query users	62.7396
Cloud	12885.2498
Trusted third party	145.0821

From the time expenditure of each stage in table 2, it is not difficult to see that the cloud has undertaken most of the calculation, which can effectively reduce the time consumption of trusted institutions, data providing users, and query users.

Table 3: Comparison between the scheme and the related literature scheme

	Secure Data Collection	Main Encryption Algorithm	User Calculation Cost	Communication Overhead	Security Query Process	Offline Query
[15]	–	Oblivious Transfer + Multiplication Triplets	Low	Higher	–	√
[19]	–	Multiplication Triplets + Paillier	Higher	Higher	√	√
[23]	√	Differential privacy + Paillier	Higher	Higher	√	–
[24]	–	SE-VHE	Low	Low	√	√
[30]	√	–	Low	Low	√	√
Ours	√	Stream cipher + GSW-FHE	Low	Lowd	√	√

As shown in table 3. In our solution, we will unload as many computing tasks as possible to the cloud by combining the computing tasks and computing power requirements of all parties. At the same time, because users store ciphertext data in the cloud, users do not need to be online or communicate with trusted institutions in real time, reducing user communication overhead.

It is first considered that the use of differential privacy and other encryption schemes during data collection will lose part of the data accuracy or bring certain computational overhead to the sensor device. Therefore, this article uses stream ciphers to transmit data. In the process of homomorphic encryption, tasks are assigned based on the computing power of each participant. At the same time, due to the large amount of data and calculation process in the calculation process, encryption methods such as Paillier may increase the communication, calculation, and storage overhead. Therefore, the GSW scheme is adopted to encrypt the data.

5.3. Extensions to other machine learning algorithms

In this section, we will discuss the scalability of the scheme and give its application in federated learning.

First of all, federated learning provides a good distributed machine learning architecture, which can ensure the localized processing of user data and data security. However, with recent research findings: the gradient passed in federated learning may also cause user data leakage. Therefore, it is necessary to design an algorithm that can effectively protect the user's local gradient. Next, we will show how to use the proposed scheme to build a secure gradient aggregation algorithm under the federated learning.

In federated learning, FedAvg is one of the very famous algorithms, and it has also received extensive attention and applications. On the basis of [44], we will improve FedAvg and complete the gradient aggregation process under ciphertext, as shown in algorithm 1-algorithm 4. This can effectively reduce the amount of user data information leaked by gradient.

First, a model gradient needs to be initialized as shown in algorithm 1.

Algorithm 1 SEMMI-FedAvg-Init

1: **for** each participant $K_i \in K$ **do**
2: Initialize model w_0
3: **end for**
4: **return** w_0;

Next, each participant needs to execute a model training algorithm with local data as shown in algorithm 2. When the gradient of the i-th round is obtained, the public key needs to be used to encrypt the gradient, and then the ciphertext is sent to the cloud, which is the algorithm 3.

After the cloud has received all the gradient information (of course, some schemes can get a good model even when there are only partial gradients), the calculation will be performed under the ciphertext. Finally, the result of the calculation is sent to a trusted organization.

The trusted organization decrypts the received calculation result and sends it to all participants.

According to algorithm 1 to algorithm 4, our scheme can complete the training process of FedAvg under ciphertext. During training, the only plaintext information sent in the network is the global gradient. But since the global gradient is a public information, this will not affect the security of the data.

6. Summary and Future Work

Compared with other application scenarios, the Internet of Medical Things has stricter requirements for data security protection. Due to the restrictions on the circulation and use of data, some applications cannot be widely used. Therefore, in order to meet the needs of data security and application scenarios. This paper proposes SEMMI. It is a secure, efficient and intelligent decision-making solution applied to the Internet of Medical Things (smart medical) scenarios. At the same time, this paper has carried out an example application in the KNN classification algorithm and proved it experimentally. In this scheme, data is carried out in the form of ciphertext during network transmission, calculation,

Algorithm 2 SEMMI-FedAvg-Local

Input: Initialize model w_0. / The return value of algorithm 4 is w_j^{Sum}.

Output: Encrypted gradient $C_{w_j^i}$.

for each round $j = 1, 2, 3, ...$ **do**
  if Execute the function for the first time **then**
    K is a randomly selected participant.
    for each participant $K_i \in K$ **do**
      $C_{w_j^i} = PubEnc(pk, w_j^i)$
    end for
    goto algorithm 3 $(C_{w_j^i}, function\ )$
  else
    K is a randomly selected participant.
    for each participant $K_i \in K$ **do**
      $w_j = \frac{w_j^{Sum}}{N}$ \\ N is the number of participants.
      $w_{j+1}^i = w_j - \alpha \mathbf{X}^T(\mathbf{X} \cdot w_j - \mathbf{Y})$
      $C_{w_j^i} = PubEnc(\mathbf{C} \leftarrow pk, w_{j+1}^i)$
    end for
    goto algorithm 3 $(C_{w_j^i}, function\ )$
  end if
end for
return ;

Algorithm 3 SEMMI-FedAvg-Cloud

Input: Encrypted gradient $C_{w_j^i}$.

Output: Ciphertext C_{w_j} after gradient aggregation.

Initialize: model C_{w_j}
for each participant $K_i \in K$ **do**
    $C_{w_j} = function(C_{w_j^i}, C_{w_j})$
    Next K_i
end for
goto algorithm 4 C_{w_j}
return ;

Algorithm 4 SEMMI-FedAvg-Server

Input: Ciphertext C_{w_j} after gradient aggregation.

Output: The return value is w_j^{Sum}.

$Dec(w_j^{Sum} \leftarrow (C_{w_j}, sk))$
return w_j^{Sum};

and storage. This will effectively protect data from being stolen or leaked by dishonest parties. All participants in the program do not need to stay online, effectively reducing the cost per participant. It can be seen from the theoretical and experimental results that this scheme meets the four design goals proposed in this paper. Finally, to demonstrate the flexibility of the scheme, we also construct an algorithm in federated learning.

This paper also discusses the more flexible application protocol process, but because the relevant encryption algorithms are currently difficult to practically apply, this is also the future research direction of this paper.

Acknowledgment

We would like to thank the anonymous reviewers for their careful reading and useful comments. This work was supported by the National Natural Science Foundation of China (62072359, 62072352, 61902292), the Key Research

and Development Programs of Shaanxi (No.2021ZDLGY06-03), and the Fundamental Research Funds for the Central Universities (No. XJS201502). We would also like to thank anonymous reviewers for their constructive comments and helpful advice.

References

[1] M. Elhoseny, G.-B. Bian, S. Lakshmanaprabu, K. Shankar, A. K. Singh, W. Wu, Effective features to classify ovarian cancer data in internet of medical things, Computer Networks 159 (2019) 147–156. doi:10.1016/j.comnet.2019.04.016.
URL https://www.sciencedirect.com/science/article/pii/S1389128618310144

[2] Y. Jin, H. Yu, Y. Zhang, N. Pan, M. Guizani, Predictive analysis in outpatients assisted by the internet of medical things, Future Generation Computer Systems 98 (2019) 219–226. doi:10.1016/j.future.2019.01.019.
URL https://www.sciencedirect.com/science/article/pii/S0167739X18327717

[3] I. U. Din, M. Guizani, J. J. Rodrigues, S. Hassan, V. V. Korotaev, Machine learning in the internet of things: Designed techniques for smart cities, Future Generation Computer Systems 100 (2019) 826–843. doi:10.1016/j.future.2019.04.017.
URL https://www.sciencedirect.com/science/article/pii/S0167739X19304030

[4] Y. Qu, N. Xiong, RFH: A resilient, fault-tolerant and high-efficient replication algorithm for distributed cloud storage, in: 41st International Conference on Parallel Processing, ICPP 2012, Pittsburgh, PA, USA, September 10-13, 2012, IEEE Computer Society, 2012, pp. 520–529. doi:10.1109/ICPP.2012.3.
URL https://doi.org/10.1109/ICPP.2012.3

[5] M. S. Hossain, G. Muhammad, Deep learning based pathology detection for smart connected healthcares, IEEE Network 34 (6) (2010) 120–125. doi:10.1109/MNET.011.2000064.

[6] G. Muhammad, M. F. Alhamid, X. Long, Computing and processing on the edge: Smart pathology detection for connected healthcare, IEEE Network 33 (6) (2019) 144–498. doi:10.1109/MNET.001.1900045.

[7] H. Jian, H. Chen, W. Xiaoyi, A smart device enabled system for autonomous fall detection and alert, International Journal of Distributed Sensor Networks 2 (12) (2016) 2308183. doi:10.1155/2016/2308183.

[8] T. Yongjun, Security design and application of internet of things based on asymmetric encryption algorithm and neural network for covid-19, Journal of Intelligent and Fuzzy Systems 39 (6) (2020) 8703–8711. doi:10.3233/JIFS-189266.

[9] W. Xing, Y. Bei, Medical health big data classification based on knn classification algorithm, IEEE Access 8 (2020) 28808–28819. doi:10.1109/ACCESS.2019.2955754.

[10] A. Gatouillat, Y. Badr, B. Massot, E. Sejdić, Internet of medical things: A review of recent contributions dealing with cyber-physical systems in medicine, IEEE Internet of Things Journal 5 (5) (2018) 3810–3822. doi:10.1109/JIOT.2018.2849014.

[11] J.-P. A. Yaacoub, M. Noura, H. N. Noura, O. Salman, E. Yaacoub, R. Couturier, A. Chehab, Securing internet of medical things systems: Limitations, issues and recommendations, Future Generation Computer Systems 105 (2020) 581–606. doi:10.1016/j.future.2019.12.028.
URL https://www.sciencedirect.com/science/article/pii/S0167739X19305680

[12] C. Lin, Y. He, N. Xiong, An energy-efficient dynamic power management in wireless sensor networks, in: 5th International Symposium on Parallel and

Distributed Computing (ISPDC 2006), 6-9 July 2006, Timisoara, Romania, IEEE Computer Society, 2006, pp. 148–154. doi:10.1109/ISPDC.2006.8.
URL https://doi.org/10.1109/ISPDC.2006.8

[13] P. VoigtAxel, A. von dem Bussche, The eu general data protection regulation (GDPR), Springer, Cham, 2017.

[14] C. Tankard, What the gdpr means for businesses, Network Security 2016 (6) (2016) 5–8. doi:10.1016/S1353-4858(16)30056-3.
URL https://www.sciencedirect.com/science/article/pii/S1353485816300563

[15] P. Mohassel, Y. Zhang, Secureml: A system for scalable privacy-preserving machine learning, in: 2017 IEEE Symposium on Security and Privacy (SP), 2017, pp. 19–38. doi:10.1109/SP.2017.12.

[16] Q. Yang, Y. Liu, T. Chen, Y. Tong, Federated machine learning: Concept and applications, ACM Trans. Intell. Syst. Technol. 10 (2). doi:10.1145/3298981.
URL https://doi.org/10.1145/3298981

[17] M. K. Hasan, S. Islam, R. Sulaiman, S. Khan, A.-H. A. Hashim, S. Habib, M. Islam, S. Alyahya, M. M. Ahmed, S. Kamil, M. A. Hassan, Lightweight encryption technique to enhance medical image security on internet of medical things applications, IEEE Access 9 (2021) 47731–47742. doi:10.1109/ACCESS.2021.3061710.

[18] R. Hamza, Z. Yan, K. Muhammad, P. Bellavista, F. Titouna, A privacy-preserving cryptosystem for iot e-healthcare, Information Sciences 527 (2020) 493–510. doi:10.1016/j.ins.2019.01.070.
URL https://www.sciencedirect.com/science/article/pii/S002002551930088X

[19] L. Liu, J. Su, X. Liu, R. Chen, K. Huang, R. H. Deng, X. Wang, Toward highly secure yet efficient knn classification scheme on outsourced cloud

data, IEEE Internet of Things Journal 6 (6) (2019) 9841–9852. doi:10.1109/JIOT.2019.2932444.

[20] J. Wang, L. Wu, H. Wang, K.-K. R. Choo, D. He, An efficient and privacy-preserving outsourced support vector machine training for internet of medical things, IEEE Internet of Things Journal 8 (1) (2021) 458–473. doi:10.1109/JIOT.2020.3004231.

[21] W. Wu, U. Parampalli, J. Liu, M. Xian, Privacy preserving k-nearest neighbor classification over encrypted database in outsourced cloud environments, World Wide Web 22 (1) (2010) 101–123. doi:10.1007/s11280-018-0539-4.

[22] B. K. Samanthula, Y. Elmehdwi, W. Jiang, k-nearest neighbor classification over semantically secure encrypted relational data, IEEE Transactions on Knowledge and Data Engineering 27 (5) (2015) 1261–1273. doi:10.1109/TKDE.2014.2364027.

[23] C. Zhou, A. Fu, S. Yu, W. Yang, H. Wang, Y. Zhang, Privacy-preserving federated learning in fog computing, IEEE Internet of Things Journal 7 (11) (2020) 10782–10793. doi:10.1109/JIOT.2020.2987958.

[24] H. Yang, S. Liang, J. Ni, H. Li, X. S. Shen, Secure and efficient ¡italic¿k¡/italic¿ nn classification for industrial internet of things, IEEE Internet of Things Journal 7 (11) (2020) 10945–10954. doi:10.1109/JIOT.2020.2992349.

[25] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, S. Moriai, Privacy-preserving deep learning via additively homomorphic encryption, IEEE Transactions on Information Forensics and Security 13 (5) (2018) 1333–1345. doi:10.1109/TIFS.2017.2787987.

[26] J. Park, D. H. Lee, Parallelly running k-nearest neighbor classification over semantically secure encrypted data in outsourced environments, IEEE Access 8 (2020) 64617–64633. doi:10.1109/ACCESS.2020.2984579.

[27] J. Liu, M. Juuti, Y. Lu, N. Asokan, Oblivious neural network predictions via minionn transformations, in: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS '17, Association for Computing Machinery, New York, NY, USA, 2017, p. 619–631. doi:10.1145/3133956.3134056.
URL https://doi.org/10.1145/3133956.3134056

[28] M. S. Riazi, C. Weinert, O. Tkachenko, E. M. Songhori, T. Schneider, 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, ASIACCS '18, Association for Computing Machinery, New York, NY, USA, 2018, p. 707–721. doi:10.1145/3196494.3196522.
URL https://doi.org/10.1145/3196494.3196522

[29] S. Wagh, D. Gupta, N. Chandran, Securenn: 3-party secure computation for neural network training, Proceedings on Privacy Enhancing Technologies 2019 (3) (2019) 26–49. doi:10.2478/popets-2019-0035.

[30] Y. Yao, N. Xiong, J. H. Park, L. Ma, J. Liu, Privacy-preserving max/min query in two-tiered wireless sensor networks, Computers and Mathematics with Applications 65 (9) (2013) 1318–1325, advanced Information Security. doi:10.1016/j.camwa.2012.02.003.
URL https://www.sciencedirect.com/science/article/pii/S0898122112001174

[31] A. Acar, H. Aksu, A. S. Uluagac, M. Conti, A survey on homomorphic encryption schemes: Theory and implementation, ACM Computing Surveys 51 (4). doi:10.1145/3214303.
URL https://doi.org/10.1145/3214303

[32] C. Gentry, A. Sahai, B. Waters, Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based, in: R. Canetti, J. A. Garay (Eds.), Advances in Cryptology -

CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I, Vol. 8042 of Lecture Notes in Computer Science, Springer, 2013, pp. 75–92. doi:10.1007/978-3-642-40041-4_5.
URL https://doi.org/10.1007/978-3-642-40041-4_5

[33] J. Alperin-Sheriff, C. Peikert, Faster bootstrapping with polynomial error, in: J. A. Garay, R. Gennaro (Eds.), Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I, Vol. 8616 of Lecture Notes in Computer Science, Springer, 2014, pp. 297–314. doi:10.1007/978-3-662-44371-2_17.
URL https://doi.org/10.1007/978-3-662-44371-2_17

[34] R. Hiromasa, M. Abe, T. Okamoto, Packing messages and optimizing bootstrapping in GSW-FHE, IEICE Trans. Fundam. Electron. Commun. Comput. Sci. 99-A (1) (2016) 73–82. doi:10.1587/transfun.E99.A.73.
URL https://doi.org/10.1587/transfun.E99.A.73

[35] H. Wang, Q. Tang, Efficient homomorphic integer polynomial evaluation based on GSW FHE, The Computer Journal 61 (4) (2018) 575–585. doi:10.1093/comjnl/bxx129.
URL https://doi.org/10.1093/comjnl/bxx129

[36] P. Mukherjee, D. Wichs, Two round multiparty computation via multi-key fhe, in: M. Fischlin, J.-S. Coron (Eds.), Advances in Cryptology – EUROCRYPT 2016, Springer Berlin Heidelberg, Berlin, Heidelberg, 2016, pp. 735–763. doi:10.1007/978-3-662-49896-5_26.

[37] P. Mukherjee, D. Wichs, Two round multiparty computation via multi-key fhe, in: M. Fischlin, J.-S. Coron (Eds.), Advances in Cryptology – EUROCRYPT 2016, Springer Berlin Heidelberg, Berlin, Heidelberg, 2016, pp. 735–763. doi:10.1007/978-3-662-49896-5_26.

[38] C. Peikert, S. Shiehian, Multi-key fhe from lwe, revisited, in: M. Hirt, A. Smith (Eds.), Theory of Cryptography, Springer Berlin Heidelberg, Berlin, Heidelberg, 2016, pp. 217–238. doi:10.1007/978-3-662-53644-5_9.

[39] Y. Chen, S. Dong, T. Li, Y. Wang, H. Zhou, Dynamic multi-key fhe in asymmetric key setting from lwe, IEEE Transactions on Information Forensics and Security 16 (2021) 5239–5249. doi:10.1109/TIFS.2021.3127023.

[40] J. S. Teh, M. Alawida, Y. C. Sii, Implementation and practical problems of chaos-based cryptography revisited, Journal of Information Security and Applications 50 (2020) 102421. doi:https://doi.org/10.1016/j.jisa.2019.102421.
URL https://www.sciencedirect.com/science/article/pii/S2214212619306544

[41] wanghs09, GSW implementation, https://github.com/wanghs09/GSW (2 2021).

[42] UCL, UCI Machine Learning Repository, http://archive.ics.uci.edu/ml/datasets.html (4 2021).

[43] Kaggle, Find Open Datasets and Machine Learning Projects, https://www.kaggle.com/datasets (5 2021).

[44] M. Naseri, J. Hayes, E. De Cristofaro, Local and central differential privacy for robustness and privacy in federated learning, in: Proceedings of the 29th Network and Distributed System Security Symposium (NDSS 2022), 2020.