2014 IEEE 14th International Conference on Bioinformatics and Bioengineering

Privacy Preserving Calculation of Fisher Criterion Score for Informative Gene Selection

Suxin Guo*, Sheng Zhong†, and Aidong Zhang‡
*‡Department of Computer Science and Engineering, State University of New York at Buffalo
†State Key Laboratory for Novel Software Technology, Nanjing University
Emails: *suxinguo@buffalo.edu, †sheng.zhong@gmail.com, and ‡azhang@buffalo.edu

***Abstract*—Informative gene selection is an important topic in the field of bioinformatics which has attracted intensive interest in recent years. It aims to identify the genes which are differentially expressed in different groups, and thus are informative for the classification between the groups. For this purpose, many microarray experiments have been conducted by various medical institutes on their own sets of patients and test subjects. For those institutes who have conducted experiments regarding the same type of disease, it would be beneficial to all of them if they learn on the union of their data to find the informative genes instead of learn just on their own datasets, since the amount of data each institute holds is very limited. However, in many cases, the institutes are not allowed to share their data with others because microarray datasets contain private information about the patients and test subjects.**

In this paper, we focus on this problem and propose a privacy preserving algorithm that allows multiple parties to perform the widely used informative gene selection method, the Fisher criterion, on the union of their data, without revealing each party's data to others. Basically, we utilize the homomorphic cryptographic system to protect the data during the calculations. Experimental results on real world datasets show the effectiveness of the proposed method.

I. INTRODUCTION

In the research of gene expression data analysis, informative gene selection has been a critical problem attracting intensive interest. Informative genes are the genes differentially expressed in different groups of samples, and are informative for the differences between those groups. They can be selected by comparing the gene expression levels between groups, and are of great value in many applications such as disease diagnosis. Some popular gene selection methods include Fisher criterion [1], correlation coefficient, which is also known as signal-to-noise statistic [2], t-test [3], and others. In this work, we focus on the Fisher criterion, which is a classical feature selection method broadly used in many areas because of its effectiveness.

With the development of the microarray analysis technology, many experiments have been conducted for the selection of informative genes regarding various kinds of diseases. These experiments are performed by medical institutes on their own sets of subjects. Due to the high cost of the DNA microarray assessment and the limited number of subjects, such experiment datasets contain only small numbers of samples, usually less than one hundred. However, there are always huge numbers of genes in the datasets compared with the numbers of samples. This fact introduces the *small sample problem*, which causes overfitting in data analysis. In such cases, the irrelevant genes might get high importance scores while the real informative genes might obtain lower importance scores due to the randomness, which reduces the performance of the gene selection [4].

A good thing is, in many cases, the microarray data regarding a particular disease is distributed among multiple institutes, because different institutes have conducted experiments regarding the same disease on different sets of subjects. This fact provides an opportunity for the institutes to collaborate with each other and learn on the union of their data so as to achieve better performance. However, the privacy concern may stop them from doing this because the microarray data contain private information about the test subjects.

To solve this problem, we propose a privacy preserving algorithm that enables multiple parties to perform the Fisher criterion method for gene selection on the union of their data without revealing each party's data to others. To the best of our knowledge, this is the first work that addresses the privacy issues in applying the Fisher criterion on distributed data. Our method can be used not only for informative gene selection, but also in other fields for feature selection with privacy concerns.

In this work, we assume that the microarray experiments and the generation of data in different institutes are performed with some common standards, so that the data from different parties are directly comparable. There are plenty of studies on the standards of microarray data, such as [5]. Since it is beyond the scope of this paper, we will not further discuss it.

The rest of this paper is organized as follows: We present the related work in **Section II** and the technical preliminaries in **Section III**. Technical preliminaries include the background knowledge about the Fisher criterion and the cryptographic tools we utilize. The details of our method is shown in **Section IV**. In **Section V**, we experimentally evaluate the algorithm. Finally, **Section VI** concludes the paper.

II. RELATED WORK

In recent years, with the growing awareness of privacy problems in data analysis, more and more data analyzing algorithms have been enhanced to be privacy preserving, including decision tree classification [6], [7], k-means clustering [8], [9], gradient descent methods [10], and others.

There are also some studies of privacy preserving feature selection and gene selection on distributed data. [11] extends the feature selection method virtual dimension reduction, which is used in the hyperspectral image processing field, to be

978-1-4799-7502-0/14 $31.00 © 2014 IEEE
DOI 10.1109/BIBE.2014.23

privacy preserving. [12] assumes the data to be distributed in a large number of parties which form an asynchronous peer-to-peer network, and develops the privacy preserving solutions for three feature selection measures in this scenario: misclassification gain, Gini index, and entropy. [13] extends the support vector machine recursive feature elimination (SVM-RFE) method to be privacy preserving on distributed data for gene selection. However, as far as we know, there is no privacy preserving solution for the Fisher criterion method, which is such a popular and effective tool for feature selection, on distributed data.

In the literature of privacy preserving data analysis, to protect the basic operations in the algorithms, many secure building blocks are developed, such as secure sum [14], secure comparison [15], [16], secure multiplication [17], secure scalar product [14], [18], [19], secure matrix multiplication [20]–[22], secure logsum computation [23], etc..

III. Technical Preliminaries

A. Fisher Criterion for Gene Selection

We briefly introduce the Fisher criterion for gene selection. Fisher criterion is a score that is calculated on each gene to evaluate how informative the gene is. The more informative a gene is, the more helpful it is in differentiating one class from another. For a gene j, its Fisher criterion score is:

$$F_j = \frac{(\mu_1^j - \mu_2^j)^2}{(\sigma_1^j)^2 + (\sigma_2^j)^2}, \tag{1}$$

where μ_1^j is the mean value of gene j for class 1 and μ_2^j is the mean value of gene j for class 2. Similarly, σ_1^j and σ_2^j are the standard deviations of gene j for classes 1 and 2, respectively.

With this criterion, the genes that obtain higher scores have mean values differ greatly between the two classes, relative to their variances [24]. Such genes are more informative in the classification.

B. Privacy Protection of the Fisher Criterion

1) Data Distribution: As explained in **Section I**, we assume that each party holds her own set of samples and the whole set of genes. This means that the data is horizontally distributed among the parties.

2) Semi-Honest Model: Our privacy preserving solution of the Fisher criterion is developed under the widely used semi-honest model [6], [10], [14], [18], [25]–[28], which assumes the parties to be "honest but curious". This means that the parties follow the protocols strictly, but they would attempt to derive the private information of other parties as much as possible from the intermediate results they obtain during the execution of the protocols. This assumption is reasonable in our case, because that all the parties are willing to obtain the correct scores, so they do not want to violate the protocols and get invalid results.

With the semi-honest assumption, we need to protect all the intermediate results during the computation, including the aggregate information such as sum and average of individual data. These information would help the parties derive the private information of others.

C. Cryptographic Tools

1) Homomorphic Cryptographic Scheme: In this work, we utilize an additive homomorphic asymmetric cryptographic system to do the encryption and decryption operations on the data. In an asymmetric cryptographic system, there is a pair of keys: a public key used for encryption and a private key used for decryption. Here we denote the encryption of a message m by $E(m)$, and the decryption of the ciphertext $E(m)$ by $D(E(m))$. Obviously, $D(E(m)) = m$.

Homomorphic cryptographic systems allow the computations to be carried out on encrypted data directly. A cryptographic system is additive homomorphic if there exist two operators $\oplus$ and $\otimes$ that, for any two integers x_1, x_2 and any constant a, we have

$$E(x_1 + x_2) = E(x_1) \oplus E(x_2),$$
$$E(a \times x_1) = a \otimes E(x_1).$$

This means, with an additive homomorphic cryptographic system, we can compute the encrypted sum of integers, $E(x_1 + x_2)$, directly from the encryptions of the two integers, $E(x_1)$ and $E(x_2)$. There is no need to decrypt them and add them up.

2) ElGamal Cryptographic system: There are several additive homomorphic cryptographic schemes [28], [29]. In this work, we use a variant of the ElGamal scheme [30]. It is semantically secure under the Diffe-Hellman Assumption [31], and has been widely used in the literature such as in [28].

ElGamal cryptographic system is a multiplicative homomorphic asymmetric cryptographic system. In this system, there are some parameters: the generator g, the private key x, and the public key y that $y = g^x$. With this system, the encryption of a message m is such a pair:

$$E(m) = (m \times y^r, g^r),$$

where r is a random integer.

We denote the first part of the pair by c_1 and the second part by c_2 so that $c_1 = m \times y^r$ and $c_2 = g^r$. To decrypt $E(m)$, we use the private key x and compute $s = c_2^x = g^{rx} = g^{xr} = y^r$. Then do $c_1 \times s^{-1} = m \times y^r \times y^{-r}$ and we can get the cleartext m.

In the variant of ElGamal scheme we use, the integer m is encrypted in such a way:

$$E(m) = (g^m \times y^r, g^r).$$

The only difference between the original ElGamal scheme and this variant is that m in the first part is changed to g^m. With the change, this variant is an additive homomorphic cryptographic system such that:

$$E(x_1 + x_2) = E(x_1) \times E(x_2),$$
$$E(a \times x_1) = E(x_1)^a.$$

To decrypt $E(m)$, we follow the same procedure as in the original ElGamal algorithm. Because of the change, after the above decryption process, we get g^m instead of m. So we add an additional step to obtain m from g^m.

In this paper, the private key is additively shared by all the parties and no party knows the complete private key. The parties have to cooperate with others to perform the decryptions, so the ciphertexts can be exposed to every party since no party can decrypt them alone.

The private key is shared in this way: Suppose there are two parties, parties A and B. A has a part of private key, x_A, and B has the other part, x_B, such that $x_A + x_B = x$, where x is the complete private key. In the decryption, we need to compute $s = c_2^x = c_2^{x_A + x_B} = c_2^{x_A} \times c_2^{x_B}$. Party A calculates $s_A = c_2^{x_A}$ and party B calculates $s_B = c_2^{x_B}$ so that $s = s_A \times s_B$. We need to do $c_1 \times s^{-1} = c_1 \times (s_A \times s_B)^{-1} = c_1 \times s_A^{-1} \times s_B^{-1}$. Party A computes $c_1 \times s_A^{-1}$ and sends it to party B. Then party B computes $c_1 \times s_A^{-1} \times s_B^{-1} = c_1 \times s^{-1} = g^m$ and sends it to A. In this way both parties can get the decrypted result. Here since party B performs her decryption part later, she gets the final result earlier. If she does not send the result to A, the decrypted result is only known to party B. The order of the parties in the decryptions can be changed, so if we need the result to be known to only one party, the party should do her decryption later.

3) Secure Multiplication: Since the additive homomorphic cryptographic system only supports the addition of integers in the ciphertext form and does not support the direct multiplication of integers in ciphertext form, we use the secure multiplication protocol that we previously proposed in [17] to achieve the multiplication operation. The input of the protocol are two encryptions of integers, $E(x)$ and $E(y)$, and the output is the encryption of the product of the two integers, $E(x \times y)$. The basic procedure of the protocol is as follows:

First, party A generates a random integer x_A. Then party A computes $E(x - x_A)$, and sends it to party B.

Second, the two parties coordinately decrypt $E(x - x_A)$ and only party B gets the result $x - x_A = x_B$.

Third, parties A and B rerandomize $E(y)$ and obtain $E'(y)$ and $E''(y)$, respectively. Then they calculate $E'(x_A \times y)$ and $E''(x_B \times y)$, respectively, and exchange the two values.

Finally, parties A and B compute $E(x \times y) = E(x_A \times y + x_B \times y)$ by themselves.

4) Secure Logsum Computation: In this work, we are inspired by the secure logsum computation proposed in [23]. The input are two d-dimensional vectors, $x = (x_1, x_2, \ldots, x_d)$, which is from party A, and $y = (y_1, y_2, \ldots, y_d)$, which is from party B. The output are two additive shares s_A held by party A and s_B held by party B that $s_A + s_B = \log(\sum_{i=1}^{d} 10^{x_i + y_i})$.

The basic idea of the secure logsum algorithm is:

First, party A computes vector 10^{x-q} where q is a random number generated by A and party B computes vector 10^y.

Second, the two parties apply the secure scalar product protocol [19] to calculate the scalar product of the two vectors 10^{x-q} and 10^y. The result $\phi = \sum_{i=1}^{d} 10^{x_i + y_i - q}$ is only known to party B.

Finally, party B computes $s_B = \log \phi = \log(\sum_{i=1}^{d} 10^{x_i + y_i}) - q$ and party A has $s_A = q$ so that $s_A + s_B = \log(\sum_{i=1}^{d} 10^{x_i + y_i})$.

IV. Privacy Preserving Calculation of the Fisher Criterion Score

In this section, we present our method to securely calculate the Fisher criterion score on data distributed between two parties A and B. It can be easily extended to the multiparty case, where the number of parties is larger than two. For simplicity, we just consider the two-party case here.

As shown in **Section III**, the Fisher criterion score of gene j is:

$$F_j = \frac{(\mu_1^j - \mu_2^j)^2}{(\sigma_1^j)^2 + (\sigma_2^j)^2},$$

where μ_1^j and μ_2^j are the mean values of gene j for classes 1 and 2, respectively, and σ_1^j and σ_2^j are the standard deviations of gene j for classes 1 and 2, respectively.

We have:

$$\mu_1^j = \frac{\sum_{x_i \in C_1} x_i^j}{n_1} = \frac{S_1^j}{n_1},$$

$$\mu_2^j = \frac{\sum_{x_i \in C_2} x_i^j}{n_2} = \frac{S_2^j}{n_2},$$

where x_i is sample i, and x_i^j is the value of sample i on gene j. $S_1 = \sum_{x_i \in C_1} x_i^j$ is the sum of all values of samples in class 1 on gene j and $S_2 = \sum_{x_i \in C_2} x_i^j$ is the sum of all values of samples in class 2 on gene j. n_1 is the number of samples in class 1, and n_2 is the number of samples in class 2.

We also have

$$(\sigma_1^j)^2 = \frac{\sum_{x_i \in C_1} (x_i^j - \mu_1^j)^2}{n_1} = \frac{\sum_{x_i \in C_1} (x_i^j - \frac{S_1^j}{n_1})^2}{n_1},$$

$$(\sigma_2^j)^2 = \frac{\sum_{x_i \in C_2} (x_i^j - \mu_2^j)^2}{n_2} = \frac{\sum_{x_i \in C_2} (x_i^j - \frac{S_2^j}{n_2})^2}{n_2}.$$

With the above calculations of means and variances, the computation of the Fisher criterion becomes

$$\begin{aligned} F_j &= \frac{(\mu_1^j - \mu_2^j)^2}{(\sigma_1^j)^2 + (\sigma_2^j)^2} \\ &= \frac{(\frac{S_1^j}{n_1} - \frac{S_2^j}{n_2})^2}{\frac{\sum_{x_i \in C_1} (x_i^j - \frac{S_1^j}{n_1})^2}{n_1} + \frac{\sum_{x_i \in C_2} (x_i^j - \frac{S_2^j}{n_2})^2}{n_2}} \\ &= \frac{n_1 n_2 (n_2 S_1^j - n_1 S_2^j)^2}{n_2^3 \sum_{x_i \in C_1} (n_1 x_i^j - S_1^j)^2 + n_1^3 \sum_{x_i \in C_2} (n_2 x_i^j - S_2^j)^2}. \end{aligned}$$

Now we show how to calculate the score securely. n_1 and n_2 are known to both parties, so we need to calculate the sums S_1^j and S_2^j, and accomplish the square and division operations securely.

A. Computation of the Sums

Here we show the computation of the encryptions of sums S_1^j and S_2^j. To compute $E(S_1^j)$, party A add up all the values of samples in class 1 on gene j in her own dataset, and obtain her own sum S_{1A}^j. Party B compute her own sum S_{1B}^j in the same way. Then they encrypt the two sums and exchange the encryptions $E(S_{1A}^j)$ and $E(S_{1B}^j)$. With the additive homomorphic property of the cryptographic system, both parties can get $E(S_1^j) = E(S_{1A}^j + S_{1B}^j)$ from $E(S_{1A}^j)$ and $E(S_{1B}^j)$. $E(S_2^j)$ can be calculated in the same way.

B. Computation of the Squares

With $E(S_1^j)$ and $E(S_2^j)$, we can compute the encryptions of the squares $(n_2S_1^j - n_1S_2^j)^2$, $\sum_{x_i \in C_1}(n_1x_i^j - S_1^j)^2$ and $\sum_{x_i \in C_2}(n_2x_i^j - S_2^j)^2$ under the help of the secure multiplication protocol [17] and the homomorphic property.

For the square $(n_2S_1^j - n_1S_2^j)^2$, $E(n_2S_1^j)$ and $E(n_1S_2^j)$ can be calculated from $E(S_1^j)$ and $E(S_2^j)$. Then $E((n_2S_1^j - n_1S_2^j)^2)$ can be calculated with the secure multiplication protocol.

To calculate the encryption of the square $\sum_{x_i \in C_1}(n_1x_i^j - S_1^j)^2$, for each sample $x_i \in C_1$, $n_1x_i^j$ is known by the party holding x_i. Then $E(n_1x_i^j)$ can be obtained. $E(S_1^j)$ is also known, so $E((n_1x_i^j - S_1^j)^2)$ can be calculated. After the squares of all the samples in class 1 are computed, the parties add up their squares and exchange their sums. Then $E(\sum_{x_i \in C_1}(n_1x_i^j - S_1^j)^2)$ can be computed. $E(\sum_{x_i \in C_2}(n_2x_i^j - S_2^j)^2)$ is obtained in the same way.

C. Computation of the Quotient

With the encryptions of the squares, we have the encryptions of $n_1n_2(n_2S_1^j - n_1S_2^j)^2$ and $n_2^3\sum_{x_i \in C_1}(n_1x_i^j - S_1^j)^2 + n_1^3\sum_{x_i \in C_2}(n_2x_i^j - S_2^j)^2$. We want to compute the quotient of them. To make it simple, we use a and b to denote the two values such that $a = n_1n_2(n_2S_1^j - n_1S_2^j)^2$ and $b = n_2^3\sum_{x_i \in C_1}(n_1x_i^j - S_1^j)^2 + n_1^3\sum_{x_i \in C_2}(n_2x_i^j - S_2^j)^2$. We have $E(a)$ and $E(b)$, and we want to compute a/b.

Directly computing a/b from $E(a)$ and $E(b)$ is hard since the additively homomorphic cryptographic system does not support the secure computation of the division operation between two encrypted integers. So we need to calculate $\log(a/b)$ instead of a/b, which turns the division into a substraction: $\log(a/b) = \log a - \log b$. Now let us show how to calculate $\log(a/b)$.

First, we change the share pattern of a and b between the parties. We want them to be additively shared between the parties in the way that party A holds a_A and b_A, and party B holds a_B and b_B, where $a_A + a_B = a$ and $b_A + b_B = b$. To achieve this, party A generates a random integer a_A and computes $E(a - a_A)$ from $E(a)$ and $E(a_A)$, and sends it to party B. Then the two parties coordinately decrypt $E(a - a_A)$ and only party B gets the result $a - a_A = a_B$. Similarly, b_A and b_B can be computed.

After this step, we can calculate $\log(a/b) = \log a - \log b = \log(a_A + a_B) - \log(b_A + b_B)$. Inspired by the secure logsum protocol [23], we calculate $\log(a_A + a_B)$ as follows:

- Party A generates a random number q, and encrypts $a_A/10^q$ and $1/10^q$. Then A sends $E(a_A/10^q)$ and $E(1/10^q)$ to B.
- After receiving $E(a_A/10^q)$ and $E(1/10^q)$, party B calculates $E(a_B/10^q)$ from $E(1/10^q)$ and her own share a_B. Then B can obtain $E(a_A/10^q + a_B/10^q) = E((a_A + a_B)/10^q) = E(10^{\log(a_A+a_B)-q})$.
- Parties A and B jointly decrypt $E(10^{\log(a_A+a_B)-q})$ and only B gets the result $\phi = 10^{\log(a_A+a_B)-q}$.
- Party B computes $\log\phi = \log(a_A + a_B) - q$. So that party A has $s_1^A = q$ and B has $s_1^B = \log(a_A + a_B) - q$. $s_1^A + s_1^B = \log(a_A + a_B)$.

Please notice that, $a_A/10^q$ and $1/10^q$ are encrypted with different random numbers, so that party B cannot derive any information by comparing the two encryptions $E(a_A/10^q)$ and $E(1/10^q)$.

Similarly, we can get $s_2^A + s_2^B = \log(b_A + b_B)$. Then we have:

$$\begin{aligned}\log(a/b) &= \log(a_A + a_B) - \log(b_A + b_B)\\ &= (s_1^A + s_1^B) - (s_2^A + s_2^B)\\ &= (s_1^A - s_2^A) + (s_1^B - s_2^B).\end{aligned}$$

$s_1^A - s_2^A$ can be computed by party A and $s_1^B - s_2^B$ can be computed by party B. The two parties then exchange the two values and both of them can get $\log(a/b)$ and calculate a/b, which is the Fisher criterion score F_j.

The whole procedure of the privacy preserving computation of the Fisher criterion score is summarized in Algorithm 1.

Algorithm 1 Privacy Preserving Calculation of the Fisher Criterion Score

```
Input: Party A and party B both have a set of samples. The total number of samples in class 1 is $n_1$ and the total number of samples in class 2 is $n_2$;
Output: The Fisher criterion score of gene $j$, $F_j$;
Party A computes her own sum of sample values on gene $j$ for the two classes, $S_{1A}^j$ and $S_{2A}^j$. So does party B;
Parties A and B exchange the encryptions of their own sums, and get $E(S_1^j)$ and $E(S_2^j)$;
The two parties compute $E((n_2S_1^j - n_1S_2^j)^2)$, $E(\sum_{x_i \in C_1}(n_1x_i^j - S_1^j)^2)$ and $E(\sum_{x_i \in C_2}(n_2x_i^j - S_2^j)^2)$ from $E(S_1^j)$ and $E(S_2^j)$ under the help of the secure multiplication protocol;
Parties A and B jointly compute the logarithm of the Fisher criterion score, $\log F_j = \log(n_1n_2(n_2S_1^j - n_1S_2^j)^2) - \log(n_2^3\sum_{x_i \in C_1}(n_1x_i^j - S_1^j)^2 + n_1^3\sum_{x_i \in C_2}(n_2x_i^j - S_2^j)^2)$;
Parties A and B compute $F_j$ from $\log F_j$;
```

D. The Practical Issue

Before we proceed to the experiment part, we first show the practical issue in implementing the algorithm. This issue appears due to the fact that the cryptographic system we use is defined over an integer ring. So we need to deal with the real numbers which are not integers.

There are two steps that would introduce such real numbers: First, the data itself may contain values which are not integers. Second, in the calculation of $\log(a_A + a_B)$, numbers like $a_A/10^q$ might not be integers.

To solve this problem, we convert the non-integer numbers to integers by multiplying them with a magnitude of 10, and then round the products to integers. After the decryption, we divide the numbers by the magnitude to recover the original numbers. Please note that we only need to do the divisions to recover the numbers produced in the logarithm computation, because the magnitude multiplied to the original data is naturally eliminated in the Fisher criterion quotient.

This is the only operation that causes accuracy loss in our method. Clearly, as the magnitudes increase, the accuracy loss should decrease. We will show this in the experimental results.

V. Experiments

In this section, we present the experimental results. Matlab is used to perform the experiments and evaluate the accuracy of our algorithm.

A. Dataset Description

We test our method on three widely used real world microarray gene expression datasets: The leukemia dataset [2], the prostate cancer dataset [32], and the SRBCT dataset [33].

The leukemia dataset is one of the most popular microarray datasets, which contains gene expression information of samples from human acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL). There are 5147 genes, and 72 samples, in which 47 are ALL samples and 25 are AML samples. All the values in this dataset are integers.

The prostate cancer dataset includes the gene expression information of samples from prostate tumors and adjacent prostate tissue not containing tumor. This dataset involves 12533 genes, and 102 samples. 50 of the samples are normal tissue samples and the other 52 samples are prostate tissue samples. The values in this dataset are also integers.

Since both of the above two datasets contain only integers, we also use the small round blue cell tumor (SRBCT) dataset, which contains non-integer real numbers, to test our method. This dataset contains four classes of childhood tumors, which are Ewing's family of tumors (EWS), neuroblastoma (NB), Burkitt's lymphoma (BL) and rhabdomyosarcoma (RMS). We use the data of the two classes: NB and BL. There are 18 NB samples and 11 BL samples. The number of genes is 2308.

B. Evaluation Metrics

Here we discuss the metrics that evaluate the accuracy of the proposed algorithm. Since our method extends the Fisher criterion to be privacy preserving, the "accuracy" in our work means the similarity between the results achieved by our method on distributed data and the results obtained with the original Fisher criterion on centralized data. The more similar the two kinds of results are, the more accurate our method is.

With the cryptographic system we use, the data are protected in the way of encryption, not perturbation. The only part that causes accuracy loss is the above mentioned rounding operations of non-integer numbers. Here this accuracy loss is measured with two metrics, the score error and the miss rate.

1) Score Error: The score error e_s is defined as the relative error between the Fisher criterion score calculated with the proposed privacy preserving approach, F^p, and the score calculated with the original approach, F^o, such that $e_s = |F^o - F^p|/F^o$. This error can be computed for each Fisher criterion score that is calculated. For each dataset and for each setting of the magnitudes of 10 used for rounding the non-integer numbers, we compute an average score error.

2) Miss Rate: Fisher criterion is used for gene selection. Genes are sorted according to their Fisher criterion scores and the genes with higher scores are considered to be more informative. We use the miss rate R_m to evaluate the miss gene selection rate of the proposed method compared with the original criterion. The miss rate is calculated as $R_m = n_e/n$, where n_e is the number of genes that are assigned different orders by the privacy preserving approach and the original approach in the sorting of Fisher criterion scores, and n is the total number of genes. For each dataset and for each setting of the magnitudes of 10 used for rounding the non-integer numbers, we compute a miss rate.

C. Experimental Results

We present the experimental results on the three datasets and show how the accuracy loss decreases as the magnitude increases.

1) Leukemia Dataset: In this dataset, since all the data values are integers, we only round the non-integer numbers generated in the logarithm computation. We show how the score error and miss rate change as the magnitude increases in Table I.

TABLE I: The Accuracy Loss of the Leukemia Dataset

Magnitude	Average Score Error	Miss Rate
10^1	1.2×10^{-7}	0
10^2	2.6×10^{-8}	0
10^3	1.6×10^{-9}	0
10^4	2.0×10^{-10}	0
10^5	1.6×10^{-11}	0

It can be found from the table that as the magnitude increases, the average score error decreases while the miss rates remains to be 0. This means that although there is certain accuracy loss in the calculation of the Fisher criterion score, this loss does not affect the selections of genes. The proposed method provides the same ordering of genes as the original criterion.

2) Prostate Cancer Dataset: Similar to the leukemia dataset, this dataset contains integers only and we only need to round the non-integer numbers generated in the logarithm computation. Table II shows the score errors and the miss rates of the proposed algorithm under different settings of the magnitude.

The change of the average score error follows the same trend as in the leukemia dataset. When the magnitude is 10^1 and 10^2, the miss rate is 1.6×10^{-4}, which means that 2 out of

TABLE II: The Accuracy Loss of the Prostate Cancer Dataset

Magnitude	Average Score Error	Miss Rate
10^1	7.9×10^{-6}	1.6×10^{-4}
10^2	1.7×10^{-6}	1.6×10^{-4}
10^3	1.1×10^{-7}	0
10^4	1.3×10^{-8}	0
10^5	1.0×10^{-9}	0

12533 genes receive different orders from the proposed method and the original method. When the magnitude increases to 10^3, the miss rate reduces to 0.

3) SRBCT Dataset: This dataset contains non-integer numbers, so we need to round both the input data values and the numbers generated in the logarithm computation. The accuracy loss is shown in Table III.

TABLE III: The Accuracy Loss of the SRBCT Dataset

Magnitude	Average Score Error	Miss Rate
10^1	1.0202	0.9809
10^2	0.0389	0.8904
10^3	0.0031	0.4749
10^4	4.1×10^{-11}	0
10^5	3.6×10^{-14}	0

It can be seen that when the magnitude is 10^1 and 10^2, the accuracy loss is huge. This is because that many data values in the dataset is around 10^{-1} and 10^{-2}, and by multiplying them with 10^1 and 10^2 and rounding them, much information is lost. For example, to round the two numbers 0.01 and 0.02 after multiplying them with 10, we can get 0 as both of the two results. However, as the magnitude increases, the accuracy loss is significantly reduced.

VI. Conclusions

In this paper, we proposed a privacy preserving algorithm that allows two parties to calculate the Fisher criterion scores of genes on the union of their samples without revealing each party's samples to the other. The calculations were achieved with the additive homomorphic cryptographic system and several secure protocols were utilized to realize the secure operations on the data. We evaluated the accuracy loss of the proposed method by comparing it with the original Fisher criterion on three real world microarray datasets and shown that with appropriate parameter setting, the accuracy loss can be very low and does not affect the selection of genes. Although we focus on the informative gene selection problem in this work, the method can be used in other feature selection problems.

References

[1] C. Bishop, "Neural networks for pattern recognition," 1995.

[2] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, and et al., "Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring," *Science*, pp. 531–537, 1999.

[3] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, "Numerical recipes in c: The art of scientific computing," pp. 504–510, 1992.

[4] K. Li, N. Du, and A. Zhang, "A link prediction based unsupervised rank aggregation algorithm for informative gene selection," *Proceedings of the 2012 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2012.

[5] J. S. J. Christian, C. C. Helen, and A. B. Catherine, "Microarray databases: standards and ontologies," *Nature Genetics 32*, pp. 469–473, 2002.

[6] Y. Lindell and B. Pinkas, "Privacy preserving data mining," *Journal of Cryptology*, vol. 15, no. 3, pp. 177–206, 2002.

[7] Z. Teng and W. Du, "A hybrid multi-group privacy-preserving approach for building decision trees," in *Proceedings of the 11th Pacific-Asia conference on Advances in knowledge discovery and data mining*, ser. PAKDD'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 296–307.

[8] J. Vaidya, W. Lafayette, and C. Clifton, "Privacy-preserving k-means clustering over vertically partitioned data," *Security*, pp. 206–215, 2003.

[9] G. Jagannathan and R. N. Wright, *Privacy-preserving distributed k-means clustering over arbitrarily partitioned data*. ACM, 2005, pp. 593–599.

[10] L. Wan, W. K. Ng, S. Han, and V. C. S. Lee, "Privacy-preservation for gradient descent methods," *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining KDD 07*, p. 775, 2007.

[11] M. Banerjee and S. Chakravarty, "Privacy preserving feature selection for distributed data using virtual dimension," *Proceedings of the 20th ACM international conference on Information and knowledge management (CIKM)*, 2011.

[12] K. Das, K. Bhaduri, and H. Kargupta, "A local asynchronous distributed privacy preserving feature selection algorithm for large peer-to-peer networks," *Knowledge and Information Systems*, 2010.

[13] F. Camara, M. L. Samb, S. Ndiaye, and Y. Slimani, "Privacy preserving rfe-svm for distributed gene selection," *International Journal of Computer Science Issues*, 2012.

[14] C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M. Y. Zhu, "Tools for privacy preserving distributed data mining," *ACM SIGKDD Explorations Newsletter*, vol. 4, no. 2, pp. 28–34, 2002.

[15] I. Damgard, M. Fitzi, E. Kiltz, J. B. Nielsen, and T. Toft, *Unconditionally Secure Constant-Rounds Multi-party Computation for Equality, Comparison, Bits and Exponentiation*. Springer, 2006, vol. 3876, pp. 285–304.

[16] I. Damgard, M. Geisler, and M. Kroigard, "Homomorphic encryption and secure comparison," *International Journal of Applied Cryptography*, vol. 1, p. 22, 2008.

[17] S. Guo, S. Zhong, and A. Zhang, "Privacy-preserving kruskal-wallis test," *Computer Methods and Programs in Biomedicine*, 2013.

[18] W. Du and M. Atallah, *Privacy-Preserving Cooperative Statistical Analysis*. IEEE Computer Society, 2001, p. 102.

[19] B. Goethals, S. Laur, H. Lipmaa, and T. Mielikainen, "On private scalar product computation for privacy-preserving data mining," *Science*, vol. 3506, pp. 104–120, 2004.

[20] S. Han and W. K. Ng, "Privacy-preserving linear fisher discriminant analysis," in *Proceedings of the 12th Pacific-Asia conference on Advances in knowledge discovery and data mining*, ser. PAKDD'08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 136–147.

[21] W. Du, Y. Y. S. Han, and S. Chen, *Privacy-preserving multivariate statistical analysis: Linear regression and classification*. Lake Buena Vista, Florida, 2004, vol. 233.

[22] S. Han, W. K. Ng, and P. S. Yu, "Privacy-preserving singular value decomposition," *2009 IEEE 25th International Conference on Data Engineering*, pp. 1267–1270, 2009.

[23] P. Smaragdis and M. Shashanka, "A framework for secure speech recognition," *IEEE Transactions On Audio Speech And Language Processing*, vol. 15, no. 4, pp. 1404–1413, 2007.

[24] H. Chai and C. Domeniconi, "An evaluation of gene selection methods for multi-class microarray data classification," *Proceedings of the Second European Workshop on Data Mining and Text Mining in Bioinformatics*, 2004.

[25] O. Goldreich, *Foundations of Cryptography*. Cambridge University Press, 2001, vol. 1, no. 3.

[26] M. Kantarcioglu and C. Clifton, "Privacy-preserving distributed mining of association rules on horizontally partitioned data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 9, pp. 1026–1037, 2004.

[27] J. Vaidya and C. Clifton, *Privacy-preserving outlier detection*. IEEE, 2004, vol. 41, no. 1, pp. 233–240.

[28] S. Zhong, "Privacy-preserving algorithms for distributed mining of

frequent itemsets," *Information Sciences*, vol. 177, no. 2, pp. 490–503, 2007.

[29] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," *Computer*, vol. 1592, pp. 223–238, 1999.

[30] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, 1985.

[31] D. Boneh, *The Decision Diffie-Hellman Problem*. Springer-Verlag, 1998, vol. 1423, pp. 48–63.

[32] D. Singh, P. G. Febbo, K. Ross, D. G. Jackson, J. Manola, C. Ladd, P. Tamayo, A. A. Renshaw, A. V. D'Amico, J. P. Richie, E. S. Lander, M. Loda, P. W. Kantoff, T. R. Golub, and W. R. Sellers, "Gene expression correlates of clinical prostate cancer behavior," *Cancer Cell*, 2002.

[33] J. Khan, J. S. Wei, M. Ringner, L. H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C. R. Antonescu, C. Peterson, and P. S. Meltzer, "Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks," *Nature Medicine*, 2001.