A Distributed Solution for Privacy Preserving Outlier Detection

Luong The Dung
Information Technology Center
Government Information Security Commission
Hanoi, Vietnam

Ho Tu Bao
School of Knowledge Science
Japan Advanced Institute of Science and Technology
Nomi City, Ishikawa, Japan

***Abstract*—In this paper, we study some parties - each has a private data set - want to conduct the outlier detection on their joint data set, but none of them want to disclose its private data to the other parties. We propose a linear transformation technique to design protocols of secure multivariate outlier detection in both horizontally and vertically distributed data models. While different from the most of previous techniques in a privacy preserving fashion for distance-based outliers detection, our focus is the technique in statistics for detecting outliers.**

I. Introduction

Nowadays, data mining is being used in various applications to support people discovering useful knowledge in large databases [1]. However, the process of mining data can result in a violation of privacy. Furthermore, issues of privacy preservation in data mining are receiving more and more attention from the this community [2]. As a result, a large number of studies has been produced on the topic of privacy-preserving data mining (PPDM) [3]. These studies deal with the problem of learning models from databases, while protecting data privacy at the level of individual records or the level of organizations.

Outlier detection is an important data mining technique that has wide application in various areas such as network intrusion detection, searching for terrorism, fraud discovery in the mobile communication, etc. The problem is that there are usually several parties participating in the mining process, each has its private data set and want to cooperate to detect outliers, but none of them wants to disclose its private data. For example, two companies need to share their network log data to build an intrusion detection system, some banks need to share their customers data to find fraud cases, etc. Therefore, some privacy preserving outlier detection methods have been developed to solve such kinds of problems [4], [5].

While there are a number of different definitions for outliers as well as techniques to find them, the existing privacy preserving methods were only proposed for Euclidean distance-based outliers detection. In addition, there are other techniques in statistics for detecting outliers [6], but still no work on this problem in a privacy preserving fashion. Basically, the statistical method requires computing the Mahalonobis distance. The secure Mahalanobis distance computation of two-party model has been proposed in [7]. However, the purpose of this work is to address the problem of privacy-preserving multivariate statistical analysis. In addition, it has only solved for the vertically distributed data on two parties. Thus, concerning to this work, our contributions in this paper are the following:

- A development of a solution for privacy preserving outlier detection in both horizontally distributed data model and vertically distributed data model on two parties.
- An extension of the proposed solution to privacy outlier detection in the horizontally distributed data model on K parties ($K > 2$).

In relation to our work, there are mainly two kinds of PPDM methods [8]: the randomization methods and the cryptographic methods. Firstly, the randomization methods randomize the original data or add noise into the original data, so the miner cannot see the original data. In the mining process, the miner has to reconstruct approximate distribution of the original data set from random values [9], [10], [11]. The randomization method has also been applied to various data mining work such as association rules [9], classification [12], privacy preserving collaborative filtering [13], etc. An important problem in randomization methods is the tradeoff between accuracy and privacy [14]. Secondly, the cryptographic methods fall under the theoretical framework of Secure Multiparty Computation [15]. These techniques allow two or many parties to cooperate for computation works on their joint data sets without disclosing each party's private information. Many cryptography techniques have been proposed for various applications [16], [17], [18]. In our work, we use some theory models defined in [15], [7].

II. Technical preliminaries

A. Problem statement

Multivariate outlier detection: Outlier detection methods can be divided into two groups of univariate methods and multivariate methods [19], [6], [1]. Multivariate outlier detection methods can be further divided into statistical methods that mainly based on estimated distribution parameters and other data mining related methods. Statistical methods for multivariate outlier detection often indicate observations that are located relatively far from the center of the data distribution. Several distance measures can be implemented for such a task. The Mahalanobis distance is a well-known criterion which depends on estimated parameters of the multivariate distribution.

Consider a data set X that has N observations and n attributes X_1, X_2, ..., X_n, where each X_i $(i = 1, 2, ..., n)$ takes values as real numbers. Denote the sample mean vector of X by $\overline{X}$, the sample covariance matrix of X by $C(X)$, and the i^{th} row of X by $X(i)$. We also define $\hat{X}$ as the following matrix $\hat{X}(i) = X(i) - \overline{X}$ and $C(X) = \frac{1}{N-1}\hat{X}^T\hat{X}$. The Mahalanobis distance d_i of the row i^{th} is defined as

$$d_i^2 = (X(i) - \overline{X})^T C^{-1}(X)(X(i) - \overline{X})$$

for $i = 1, ..., N$. Usually, $\overline{X}$ is the multivariate arithmetic mean vector. A large distance indicates that observation is an outlier for predictor. Therefore, in order to detect outliers, the main task is to effectively compute $C^{-1}(X)$ and $\overline{X}$.

Privacy preserving problem for outlier detection: Assuming that X is horizontally or vertically distributed on K parties. The target of this work is to find solutions to conduct multivariate outlier detection on the joint data set of all parties. So each party can know which object is an outlier without disclosing information about the objects and the local statistical parameters of each party.

B. Linear transformation

Let M be a $n \times n$ invertible matrix, each m_{ij} entry of M takes a random value as a real number, and $Y = XM$ is a random $N \times n$ matrix obtained from a linear transformation of X somewhere they said that matrix M.

Lemma 1. *Let $G(X)$ be the gram matrix of X, that is, $G(X) = X^T X$. Then, the gram matrix of Y will be given by $G(Y) = M^T G(X) M$ and the inverse matrix of $G(X)$ will be given by $G^{-1}(X) = MG^{-1}(Y)M^T$*

Proof: We consider each entry of G(Y):

$$G_{ij}(Y) = Y_i^T Y_j = \sum_{s=1}^{n} m_{si} X_s \sum_{t=1}^{n} m_{tj} X_t$$

$$= \sum_{s=1}^{n} \sum_{t=1}^{n} m_{si} m_{tj} X_s X_t$$

$$= \sum_{s=1}^{n} \sum_{t=1}^{n} m_{si} m_{tj} G_{st}(X)$$

Thus, $G(Y) = M^T G(X) M$, where X_i and Y_i are the column vectors of X and Y, respectively.

In addition, for any two invertible square matrices P and Q, we have $(PQ)^{-1} = Q^{-1}P^{-1}$. So, using this equation,

$$\begin{aligned} G^{-1}(Y) &= (M^T G(X) M)^{-1} \\ &= ((M^T G(X))M)^{-1} \\ &= M^{-1}(M^T G(X))^{-1} \\ &= M^{-1} G(X)^{-1} (M^T)^{-1} \end{aligned}$$

and then $G^{-1}(X) = MG^{-1}(Y)M^T$

■

C. Privacy model

In some early studies in secure multi-party computation, any computation of a party participating in protocol can only be computed based on the party's input and output. So, each party only has the access right to its input and output, and no additional information is learned. This security property is very useful because it does not disclose extra information; however, it is difficult to achieve efficiently.

In order to achieve the cost of communication more efficiency, some works have extended SMC to more complicated computation circumstances. In [7], the authors proposed a new security model for secure two-party computation.

Definition 1. *(Expansion security model) Assume all input are in the real domain R. Denote I_i be the private input of each P_i and O_i the output of P_i $(i = 1, 2)$. Let C presents the computation between two parties, i.e. $(O_1, O_2) = C(I_1, I_2)$. A protocol C is secure against dishonest P_1 if there exists an infinite number of (I'_2, O'_2) pairs in (R, R) such as $(O_1, O'_2) = C(I_1, I'_2)$. Similarly, the protocol C is secure against dishonest P_2 if there exists an infinite number of (I'_1, O'_1) pairs in (R, R) such that $(O'_1, O_2) = C(I'_1, I_2)$.*

This model is weaker in secure than the SMC security model. Currently, it is still considered as a heuristic model and theoretically, analysis of this model is still being investigated. However, using this model can lead to solutions that are much more efficient than the solutions based on the SMC security model. Theoretically, a protocol that satisfies the K-SMC model might still disclose significant information. However, it doesn't happen in the situations applied in this paper.

D. Private matrix product sharing

In this paper, we also use the private matrix product sharing protocol proposed in [7] as a building block to incorporating privacy preservation in the next protocols.

Let $A = (a_{ij})_{m\times q}$ and $B = (b_{ij})_{q\times n}$ be two private matrices of the parties Alice and Bob, respectively. The goal of this protocol is to privately compute the product AB in which Alice and Bob obtain the random matrices S^a and S^b respectively, where $S^a + S^b = AB$.

E. Secret mean sharing

Assume that two parties Alice and Bob want to share a value z, in such a way that Alice holds (x, n), Bob holds (y, m), and z is equal to (x + y)/(m+n). This is called secret mean sharing. The result of sharing allows Alice and Bob to obtain the random values r_A and r_B, respectively where $r_A + r_A = z$. The protocol for this problem was described in [20].

III. Protocols for the horizontally distributed data

The data set X is horizontally distributed on K parties, each party P_k $(k = 1, 2, ..., K)$ has a subset X^k with N_k observations and all of n attributes. In other words, each P_k $(k = 1, 2, ..., K)$ can only observe N_k observations of the set of N observations.

Let $X(j)$ presents the j^{th} row of the matrix X^k and $\hat{X}^k$ is the matrix obtained by $\hat{X}^k = X(j) - \overline{X}$. Let G and $G^{(i)}$ be gram matrices of $\hat{X}$ and $\hat{X}^i$, respectively. That is, $G = \hat{X}^T\hat{X}$ and $G_k = (\hat{X}^k)^T\hat{X}^k$. Then

$$G = \sum_{k=1}^{K} G_k = \sum_{k=1}^{K} (\hat{X}^k)^T \hat{X}^k$$

From this property, we can present $C(X)$ as follows:

$$C(X) = \frac{1}{N-1}\sum_{k=1}^{K} G^{(k)} = \sum_{i=1}^{K} C^{(k)}$$

Each element C_{ij} of $C(X)$ is computed by

$$C_{ij} = \frac{g_{ij}}{N-1} = \frac{\sum_{k=1}^{K} g_{ij}^{(k)}}{\sum_{k=1}^{K} N_k - 1}$$

where $g_{ij}^{(k)}$ is an element of matrix $G^{(k)}$ $(1 \leq i, j \leq n)$ that owned by the party k.

In the case with only two parties, Alice and Bob. We have

$$C_{ij} = \frac{g_{ij}}{N-1} = \frac{g_{ij}^{(1)} + g_{ij}^{(2)}}{N_1 + N_2 - 1}$$

where $g_{ij}^{(1)}$ and $g_{ij}^{(2)}$ are elements of matrices $G^{(1)}$ (owned by Alice) and $G^{(2)}$ (owned by Alice), respectively.

In the next sections, we present two protocols for privacy preserving outlier detection in horizontally distributed data. Two important computation works need to be implemented that compute $\overline{X}$ and $C^{-1}(X)$. Here $\overline{X}$ can be directly obtained by the multi-party division protocol without disclosing raw data of each party. Consequently, basically, the main work is to compute $C^{-1}(X)$.

A. Two-party protocol

In this section, we introduce a protocol for the two-party model. Our protocol consists of four computation works. Firstly, two parties involve in the secure mean sharing protocol to compute $\overline{X}$. Secondly, the parties privately shares the matrix $C(X)$ using the private matrix product sharing (PMPS) protocol [7]. For each element C_{ij} of $C(X)$, parties implement secure mean sharing protocol. So, Alice obtains $C_{ij}^{(1)}$ and Bob obtains $C_{ij}^{(2)}$, where $C_{ij}^{(1)} + C_{ij}^{(2)} = C_{ij}$. At the end of this computation work, Alice obtains a random matrix $C^{(1)}$ and Bob obtains $C^{(2)}$, where $C^{(1)} + C^{(2)} = C(X)$. Thirdly, two parties involve in three steps to compute the inverse covariance matrix by using the linear transformation method together with the PMPS protocol. The fourth computation is to locally obtain the Mahalonobise distance of every object for each party. The more detail of this protocol is given in Figure 1.

Analysis of Protocol: Based on Lemma 1, it is easy to prove that the protocol is correct. Thus, here we only consider its privacy.

In our protocol, two parities are assumed to be semi-honest who strictly follow the protocol but collect all intermediate results during the execution of protocols to learn the private data of the other party. As we observe, this protocol applies two main secure building blocks: the private matrix product sharing protocol and secure mean sharing protocol. First block depends on the expansion privacy model that is provably privacy [7]. The security of second block was proved based on Semi-honest model. Based on random share technique, we actually split all the intermediate results into two random shares except the inverse matrix covariance $C^{-1}(X)$, the mean vector $\overline{X}$ and the product matrix $C(Y) = C(X)M$ (revealed to Bob). For random shares, the private variables of one party are protected by the equivalent numbers of random portions known by itself only. Therefore we can obtain data privacy of honest parties. In addition, the revealing $\overline{X}$ and $C^{-1}(X)$ does not pose any privacy information. For $C(Y) = C(X)M$, if M is a random and non-singular matrix, $C(X)$ will be hidden in it. Theoretically, since each element o of $C(X)$ and M are in the real domain, therefore it is possible existing an infinite number of $(C(X), M)$ pairs that satisfy $C(Y) = C(X)M$. So Bob can not derive raw data of $C(X)$, this property meets the requirement of security model defined in the previous section. However, there are several attack problems that is similar to the problems analyzed in the random rotation perturbation approach [21].

Input: Alice and Bob have the data set $X^{(1)}$ and $X^{(2)}$, respectively

Output: The Mahalanobise distance of each object.

1) The parties use the secure mean sharing protocol to compute $\overline{X}$.
2) The parties share the matrix $C(X)$ by using the secure mean sharing protocol. Alice obtains $C^{(1)}$ and Bob obtains $C^{(2)}$.
3) Alice generates a random matrix M. Alice and Bob use PMPS to share $C^{(2)}M$, Alice obtains $M^{(1)}$ and Bob obtains $M^{(2)}$.
4) Alice sends $C^{(1)}M + M^{(1)}$ to Bob
5) Bob computes $C(Y) = C^{(1)}M + M^{(1)} + M^{(2)}$, then computes $C^{-1}(Y)$ and sends it to Alice.
6) Alice computes $C^{-1}(X) = MC^{-1}(Y)M^T$
7) Each party uses $C^{-1}(X)$ and $\overline{X}$ to locally compute Mahalonobise distance for its every object.

Fig. 1. Protocol for two-party horizontally distributed data.

The Independent Component Analysis (ICA) could be considered as the most commonly used method to cause the privacy breaches to the our transformation approach. ICA is a fundamental problem in signal processing which is used for blind source separation of mixed signals. Let matrix X composes by the source signals, where each row vector is a signal. Suppose we can observe the mixed signals Y, which is generated by linear transformation $C(Y) = C(X)M$. Using the ICA model can estimate the row vectors of the original signals $C(X)$, from the mixed signals $C(Y)$. To address this problem, we carefully select the transformation matrix such that the chosen perturbation is more resilient to the ICA-based attacks. Methods for selecting the transformation matrix is being under investigation.

Some other attacks to the data perturbation techniques such as approximately reconstructing to estimating the original data, estimating the properties of the values based on the distributions of the original columns are known, etc. The metric is used to measure the robustness of the perturbation technique, which is the width of the estimation range $2c\delta$ where c is a constant depending on the distribution of $\triangle D$ and the confidence level, $\triangle D = C(Y_i) - C(X_i)$ presents a random vector that is the difference between a original data column and a perturbation data column, δ is the standard deviation of $\triangle D$.

Complexity estimation: The main complexity of this protocol is derived the computational complexity of two protocols: the private matrix product sharing and the secure mean sharing. As in Steps 1 to 2, the secure mean sharing product protocol is invoked once to compute the vector $\overline{X}$ and $C(X)$. There are n elements in $\overline{X}$ and n^2 elements in $C(X)$. Therefore, the secure mean sharing protocol is invoked $n+n^2$ times in Steps 1 to 2 for computing $\overline{X}$ and C, which run $3(n+n^2)$ times of the OPE for the polynomial of degree 1. In Step 3, the private matrix product sharing protocol is invoked once for splitting the $C^{(2)}M$. It requires to use $O(n^3)$ multiplications and additions. Therefore, the overall computational complexity are $O(n^2)$ as the computational complexity of the OPE for the polynomial with degree of 1, and $O(n^3)$ multiplications and $O(n^3)$ additions.

The communication between two parties mainly comes from depends on secure mean sharing and private matrix sharing. Based on the analysis above, the communication complexity is $O(Tn^2 + Fn^3)$ bits, where T is the size for the security parameters of the obvious polynomial evaluation for the polynomial protocol (about 1024), and F is the size of the real numbers (about 32).

B. Multi-party protocol

This section, we extend the two-party protocol for the K-party case. In this protocol, one party plays the role as a master (e.g., Party 1) for initializing the protocol. Our protocol consists of the following computations. Firstly, the parties involve in the secure sum protocol to compute $N = \sum_{i=1}^{K} N_i$, next each party locally the matrix $C^{(k)}$, where each element C_{ij} of $C^{(k)}$ is computed by $C_{ij}^{(k)} = g_{ij}^{(k)}/(N-1)$. Secondly, the parties compute the inverse covariance matrix by using the linear transformation method, secure sum computation, and the PMPS protocol. The fourth computation is to locally obtain the Mahalonobise distance of every object for each party. The detail protocol given is in Figure 2

Input: K parties, each party i has the data set $X^{(i)}$.

Output: The Mahalanobise distance of each object.

1) The parties use the secure sum protocol to compute $N = \sum_{i=1}^{K} N_i$
2) Each party locally the matrix C^k.
3) Party 1 generates a random matrix M, then each party i $(i = 2, ..., K)$ and party 1 uses the private matrix product sharing protocol to share $C^{(i)}M$, party 1 obtains $M_i^{(1)}$ and party i obtains $M_i^{(2)}$.
4) Party 1 computes $C^{(1)}M + \sum_{i=1}^{K} M_i^{(1)}$, then the parties follows a communication round to compute $C(Y) = \sum_{i=1}^{K} C^{(i)}M$. At the end, Party K obtains $C(Y)$.
5) Party K computes $C^{-1}(Y)$ and sends it to Party 1.
6) Party 1 computes $C^{-1}(X) = MC^{-1}(Y)M^T$ and broadcasts this matrix to all other parties.
7) Each party uses $C^{-1}(X)$ and $\overline{X}$ to locally compute the Mahalonobise distance for its every object.

Fig. 2. Protocol for multi-party horizontally distributed data.

Analysis of protocol: Our solution preserves privacy. Indeed, privacy at the step 1 is derived from the secure sum protocol, and the privacy at the steps 3, 4, and 5 is derived from the two-party protocol. In addition, the local computations at steps 2, 6 and 7 are directly computed from the local data. However, we need at least three participating parties, each party gets result of final global computation that is sum of results of local computations. Because of only two parties participating in the computation, a party can be get the local matrix of other party by subtracting its local matrix.

Complexity estimation: Basically, the protocol complexity is bounded by the private matrix product sharing, but it is more expensive than K times in comparison with the two-party protocol. Thus, its computational complexity is $O(Kn^3)$ multiplications and $O(Kn^3)$ additions and the communication complexity $O(FKn^3)$ bits. Note that the secure mean sharing protocol used in the previous protocol are replaced by the secure sum computation. The complexity of secure sum computation is quite small in comparison with the private matrix product sharing. Thus we avoid it in this estimation.

IV. Protocol for two-party vertically distributed data

We introduce a solution for two-party vertically distributed data model. Assume that the data set X is vertically distributed on two parties Alice and Bob, where Alice has a subset X^1 and Bob has a subset X^2. In order to detect outliers, the first work is to compute $C^{-1}(X)$ and $\overline{X}$. Secondly, we use these parameters to computing Mahalonobise distance for outlier detection. $\overline{X}$ can be directly obtained by local computation, so it does not disclose raw data while computing $\overline{X}$.

For first work, we present the way to compute $C^{-1}(X)$ without disclosing the raw data as follows. In order to compute $C^{-1}(X)$, we need to compute $\hat{X}^T\hat{X}$. Since

$$\hat{X}^T\hat{X} = (\hat{X}^1 : \hat{X}^2)^T(\hat{X}^1 : \hat{X}^2)$$

$$= \begin{pmatrix} (\hat{X}^1)^T\hat{X}^1 & (\hat{X}^1)^T\hat{X}^2 \\ (\hat{X}^2)^T\hat{X}^1 & (\hat{X}^2)^T\hat{X}^2 \end{pmatrix}$$

Note that $(\hat{X}^2)^T\hat{X}^2$ and $(\hat{X}^1)^T\hat{X}^1$ can be locally computed without disclosing raw data. In addition, using private matrix sharing protocol, Alice can obtains A_1 and A_2, Bob can obtains B_1 and B_2, where $A_1 + B_1 = (\hat{X}^1)^T\hat{X}^2$ and $A_2 + B_2 = (\hat{X}^2)^T\hat{X}^1$. Therefore, Alice and Bob can obtain $C^{(1)}$ and $C^{(2)}$, respectively, where $C^{(1)} + C^{(2)} = \hat{X}^T\hat{X}$, and $C^{(1)}$ and $C^{(2)}$ can be presented as follows

$$C^{(1)} = \begin{pmatrix} (\hat{X}^1)^T\hat{X}^1 & A_1 \\ A_2 & 0 \end{pmatrix}$$

$$C^{(2)} = \begin{pmatrix} 0 & B_1 \\ B_2 & (\hat{X}^2)^T\hat{X}^2 \end{pmatrix}$$

next we uses the linear transformation method to obtain the inverse of the covariance matrix as steps 3, 4 and 5 in Figure 1.

For second work, two parties cooperatively compute the Mahalonobise distance for each object i: $d_i = (X(i) - \overline{X})^T C^{-1}(X)(X(i) - \overline{X})$. In vertically partitioned data, we assume the first n_1 dimensions of data vector $X(i) - \overline{X} = [x_1, x_2, ..., x_n]$ are held by Alice: $x_a = [x_1, x_2, ..., x_{n_1}]$ and the remaining n_2 dimensions are held Bob: $x_b = [x_{n_1+1}, x_{n_1+2}, ..., x_{n_1+n_2}]$. Assume each column vector j of the matrix C^{-1} is divided into two portions: $c_j^1 = [c_j^1, c_j^2, ..., c_j^{n_1}]^T$ and $c_j^2 = [c_j^{n_1+1}, c_j^{n_1+2}, ..., c_j^{n_1+n_2}]^T$. Let $V_a = [a_{n_1+1}, ... + a_{n_1+n_2}]$ and $V_b = [b_1, ... + b_{n_1}]$ where $a_j = x_a c_j^1$ and $b_j = x_b c_j^2$. The distance d_i can be computed by

$$\begin{aligned} d_i &= \sum_{j=1}^{n}(x_a c_j^1 + x_b c_j^2)x_j \\ &= \overbrace{\sum_{j=1}^{n_1} a_j x_j}^{r_a} + \sum_{j=n_1+1}^{n_1+n_2} a_j x_j + \sum_{j=1}^{n_1} b_j x_j + \overbrace{\sum_{j=n_1+1}^{n_1+n_2} b_j x_j}^{r_b} \end{aligned}$$

Thus, $d_i = r_a + V_a x_b^T + V_b x_a^T + r_b$. To compute d_i, parties locally compute r_a and r_b, and then they compute $V_a x_b^T$ and $V_b x_a^T$ by using the product scalar product protocol. We describe steps for computation in Figure 3

Analysis of protocol: Based on Lemma 1, it is easy to prove that this protocol allows Alice and Bob to obtain the Mahalonobise distance for their every object.

In order to analyze privacy of this protocol, we need to find out how much Alice and Bob know about each others' information at each step. At step 1 Alice and Bob using the PMPS protocol that it actually splits all the intermediate results into two random portions except the matrix $C^{-1}(X)$. The private variables of one party are protected by the equivalent numbers of random portions known by itself only. Therefore data privacy of honest parties are preserved. In the step 2, Alice only sends $C^{(1)}M + M^{(1)}$ to Bob, this matrix is only the random matrix, so it does not disclose any significant information of Alice to Bob. The disclosed information at steps 3, 4 and 5 is similar to which of the protocol for two-party horizontally distributed data, the privacy property of these steps obtains from the expansion security model. The security of the final step is based on the scalar product protocol.

Input: Alice and Bob have the data set $X^{(1)}$ and $X^{(2)}$, respectively

Output: The Mahalanobise distance of each object.

1) Alice and Bob jointly share the $C(X)$ using PPMS protocol. Alice obtains the matrix $C^{(1)}$ and Bob obtains $C^{(2)}$
2) Alice generates a random matrix M. Alice and Bob use PMPS to share $C^{(2)}M$, Alice obtains $M^{(1)}$ and Bob obtains $M^{(2)}$.
3) Alice sends $C^{(1)}M + M^{(1)}$ to Bob
4) Bob computes $C(Y) = C^{(1)}M + M^{(1)} + M^{(2)}$, then computes $C^{-1}(Y)$ and sends it to Alice.
5) Alice computes $C^{-1}(X) = MC^{-1}(Y)M^T$
6) Two parties uses the secure scalar product protocol to compute Mahalonobise distance for its every object.

Fig. 3. Protocol for two-party vertically distributed data.

The complexity estimation: The main complexity of this protocol is derived the computational complexity of two protocols: PMPS and SSP. Thus, it requires to use $O(Nn)$ multiplications, $O(Nn)$ additions, and $O(N)$ as the computational complexity of The SSP protocol with the vector length n.

Also, the communication between two parties mainly comes from depends on SSP computation and private matrix sharing. Based on the analysis above, the the communication complexity is $O((S + F)Nn)$ bits, where S is the size for the security parameters of the SSP protocol (about 1024), and F is much smaller than S (about 32), thus $O(SNn)$ bits are the communication complexity of the protocol.

V. Experiments

In this section, we provide an experiment to evaluate the performance of the proposed protocols. Protocols run in the C# language of Microsoft Visual Studio 2005 environment. All experiments are performed on the Window XP operating system with Intel core 2 duo E7500 2.93GHz and 2GB memory. As communication complexity depends on the network performance and physical distance of two parties, we simply considered parties as threads that exchange data directly by shared memory method.

The dataset used is the Breast Cancer Database from the UCI Machine Learning Depository. There are 569 data samples and 32 numeric attributes. We only use 500 data samples and 20 attributes for our experiments. We evaluate the performance of protocols for two-party distributed data. We use the OPE protocol in [22] and the SSP protocol in [23] for these experiments

Table I illustrates our measurements of the computation time for horizontally distributed data, where data instances of data set are uniformly distributed between two parties: it is linear in N, and dependencies on n is very negligible.

	N				
n	100	200	300	400	500
3	0.55	0.57	0.58	0.58	0.61
5	1.23	1.26	1.37	1.39	1.41
7	2.39	2.43	2.45	2.53	2.57
10	5.15	5.15	5.42	5.58	5.66
20	10.24	10.27	10.35	10.35	10.47

TABLE I
THE PARTIES'S COMPUTATIONAL TIME FOR THE HORIZONTALLY DISTRIBUTED DATA

	N				
n	100	200	300	400	500
3	1.49	2.41	4.14	5.08	5.93
5	2.11	5.26	6.50	7.33	8.61
7	3.66	6.01	6.49	7.84	10.13
10	4.69	6.70	10.93	14.96	19.40
10	9.31	12.76	20.15	25.47	36.40

TABLE II
THE PARTIES'S COMPUTATIONAL TIME FOR THE VERTICALLY DISTRIBUTED DATA

For a typical scenario where $n = 20$ and $N = 500$, the computation time of the protocol is about 10.47 seconds. Table II illustrates our measurements of the computation time for vertically distributed data, where attributes of data set are uniformly distributed between two parties: it is linear in both N and n. For a typical scenario where $N = 500$ and $n = 20$, the computation time of the miner is about 36.40 seconds. We can see that our method is efficient for horizontally distributed data.

VI. CONCLUSIONS

We have proposed a solution for privacy-preserving multivariate outlier detection in both vertically and horizontally distributed data model on two parties. We have extended the solution for K-party horizontally distributed model. Our solution is based on the following techniques: linear transformation, private matrix product sharing, secure mean computation and secure sum. We proved protocols's privacy based on both Semi-honest and expansion security models. We provided the experiments to show that the complexity of our protocol is linear in the number of data attributes and the size of database. In the model of horizontally distributed data, since the complexity of our protocol mainly depends on the number of attributes, it is very efficient. Our solution allows K parties to cooperate for outlier detection on their joint data sets without disclosing each party's private data to the other parties. For the future work, we will use these solutions to do some selected real-life applications.

REFERENCES

[1] J. Han and M. Kamber, *Data Mining: Concepts and Techniques 2nd ed (The Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann Publishers, 2006.

[2] T. E. Parliament, "Eu directive 95/46/ec of the european parliament and of the council on the protection of individuals with regard to the processing of personal data and on the free movement of such data," *Official J. European Communities*, vol. 40, p. 31, 1995.

[3] V. S. Verykios, E. Bertino, I. N. Fovino, L. P. Provenza, Y. Saygin, and Y. Theodoridis, "State-of-the-art in privacy preserving data mining," *SIGMOD Rec.*, vol. 33, no. 1, pp. 50–57, 2004.

[4] J. Vaidya and C. Clifton, "Privacy-preserving outlier detection," in *Proceedings of the Fourth IEEE International Conference on Data Mining*, ser. ICDM '04. IEEE Computer Society, 2004, pp. 233–240.

[5] A. Xue, X. Duan, H. Ma, W. Chen, and S. Ju, "Privacy preserving spatial outlier detection," in *Proceedings of the 2008 The 9th International Conference for Young Computer Scientists*. IEEE Computer Society, 2008, pp. 714–719.

[6] V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artif. Intell. Rev.*, vol. 22, pp. 85–126, 2004.

[7] W. Du, S. Chen, and Y. S. Han, "Privacy-preserving multivariate statistical analysis: Linear regression and classification," in *In Proceedings of the 4th SIAM International Conference on Data Mining*, 2004, pp. 222–233.

[8] A. C. Charu and P. S. Yu, *Privacy-Preserving Data Mining: Models and Algorithms*. Boston, MA, United States: ASPVU, 2008.

[9] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke, "Privacy preserving mining of association rules," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2002, pp. 217–228.

[10] S. Agrawal and J. R. Haritsa, "A framework for high-accuracy privacy-preserving mining," in *ICDE '05: Proceedings of the 21st International Conference on Data Engineering*. IEEE Computer Society, 2005, pp. 193–204.

[11] W. Du and Z. Zhan, "Using randomized response techniques for privacy-preserving data mining," in *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2003, pp. 505–510.

[12] R. Agrawal and R. Srikant, "Privacy-preserving data mining," *SIGMOD Rec.*, vol. 29, no. 2, pp. 439–450, 2000.

[13] H. Polat and W. Du, "Privacy-preserving collaborative filtering using randomized perturbation techniques," in *ICDM '03: Proceedings of the Third IEEE International Conference on Data Mining*. IEEE Computer Society, 2003, p. 625.

[14] A. Evfimievski, J. Gehrke, and R. Srikant, "Limiting privacy breaches in privacy preserving data mining," in *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. ACM, 2003, pp. 211–222.

[15] O. Goldreich, *The Foundations of Cryptography, volume 2, chapter 7: General Cryptographic Protocols*, 2nd ed. Cambridge University Press, 2004.

[16] J. Vaidya, M. Kantarcziouglu, and C. Clifton, "Privacy-preserving naive bayes classification," *The VLDB Journal*, vol. 17, no. 4, pp. 879–898, 2008.

[17] J. Vaidya and C. Clifton, "Privacy preserving association rule mining in vertically partitioned data," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2002.

[18] W. Du and Z. Zhan, "Building decision tree classifier on private data," in *Proceedings of the IEEE international conference on Privacy, security and data mining*. Australian Computer Society, Inc., 2002, pp. 1–8.

[19] F. A. Alqallaf, K. P. Konis, R. D. Martin, and R. H. Zamar, "Scalable robust covariance and correlation estimates for data mining," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '02. ACM, 2002, pp. 14–23.

[20] L. T. Dung and H. T. Bao, "Privacy preserving em-based clustering," in *IEEE RIVF International Conference on Computing and Communication Technologies(RIVF09)*. IEEE Express, 2009, pp. 111–117.

[21] C. Keke and L. Ling, "Privacy preserving data classification with rotation perturbation," in *Proceedings of the Fifth IEEE International Conference on Data Mining*, ser. ICDM '05. IEEE Computer Society, 2005, pp. 589–592.

[22] M. Naor and B. Pinkas, "Efficient oblivious transfer protocols," in *SODA '01: Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2001, pp. 448–457.

[23] B. Goethals, S. Laur, H. Lipmaa, and T. Mielikinen, "On private scalar product computation for privacy-preserving data mining," in *In Proceedings of the 7th Annual International Conference in Information Security and Cryptology*. Springer-Verlag, 2004, pp. 104–120.