International Journal of Management, IT & Engineering
Vol. 14 Issue 12, December 2024,
ISSN: 2249-0558 Impact Factor: 7.119
Journal Homepage: http://www.ijmra.us, Email: editorijmie@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gate as well as in Cabell's Directories of Publishing Opportunities, U.S.A

Privacy Preservation in Outlier Detection

Ajinkya Rasam
Aditya Sawant
Rushmere Fernandes
Varun Brahmkshatriya

Abstract (12pt)

This report presents an investigation of different privacy protection techniques used in detection of outliers. Outlier Detection is the primary step in data mining applications. The basic idea is to distinguish between data in the area of interest and abnormalities in the dataset. However, differential privacy demands indistinguishably i.e. the presence or absence of any instance in the database should be concealed such that there should not be a way to distinguish two objects from one another. Thus, the requirements of outlier detection and that of differential privacy contradict with each other. Many techniques have been developed to address this problem and our aim is to discuss various approaches taken to tackle this problem. The main idea behind making of this report is to make available all the research done and techniques developed in maintaining privacy while detecting outliers at one place.

Keywords:

Outlier Detection;
Differential Privacy;
LOF

Author correspondence:

Ajinkya Rasam,
Senior Solutions Architect, Google Cloud.
Masters in information Technology, Rutgers Business School,
IEEE Senior member,
San Francisco, California, USA
Email: rasam.ajinkya92@gmail.com

1. Introduction

Outlier detection is used in various fields and can be a useful technique in detecting credit card frauds, detecting anomalies in the field of ecommerce as well as in national defense. The application of this technique ranges from detecting unusual deductions in a bank statement to identifying suspicious terrorist activities. Thus, it is essential to develop more accurate and robust algorithms which can be applied on large databases and which will yield precise results. However, while trying to achieve this there is one thing which should not be neglected and that is „privacy". It is required to maintain the standards of differential privacy while attaining this goal.

Privacy preserving outlier detection will ensure that these concerns are balanced which will allow us to perform outlier detection task without worrying about compromising the privacy of any instance of the database [1]. Due to a wide range of applications environments there are mainly five techniques which are used to perform outlier detection task which are distance based [1], statistical based [2] , deviation based [3], density based [4] and depth based [5]. In this paper we will discuss these approaches in detail, the algorithms which are proposed for these approaches and their advantages along with their shortcomings.

There are several ideas proposed to understand the notion of distance based outlier detection approach. However, the basic idea states that an object is an outlier if it is distant from the other points in a group. One of the easiest ways to detect an outlier used by a distance based approach is to measure the distance of an

object from its k – nearest neighbors. The outlier score can be assigned to each instance or an object and it varies from zero for nearest (same) to infinity for the most distant object.

Statistical approach of outlier detection is based upon a model. A model is designed for a particular database and then it is evaluated based on its performance. The basic idea is to evaluate how will the designed model fits the data. Most of such models uses probability distribution of and evaluates the likelihood of data being fitted into that model. In this way the probability of every instance in the data can be estimated.

In density based approach to detect an outlier, an instance is treated as an anomaly if it is in the region of low density. This approach is used in measuring local outlier factor (LOF) [4] wherein each object is assigned a degree of being an outlier. It is the degree of isolation of a particular object with respect to its surrounding neighbors.

Thus, we need to discuss in detail at least one example of how privacy is protected while detecting an outlier based on each of these approaches.

2. Results and Analysis (12pt)

In this section we are going to discuss works of various authors in the field of data privacy. We aim to illustrate the basic idea behind each approach and refer the publication in the reference section for further investigation. First, we start with distance based approach and discuss all the techniques stated in „*Privacy – Preserving Outlier Detection'* [1].

2.1 Privacy preservation in Distance Based Approach

In this paper it is assumed that the data is distributed amongst different parties and only stewards (curator) are allowed to access the actual data. The problem here is to find the outlier without disclosing any further information about the outlier such that even which object an outlier need not been known to all the parties.

Usually the data can be distributed either horizontally or vertically and depending upon the way the data is distributed, solutions differ. In either case there are:

k – Participating parties ranging from P0 , . . . , P$k-1$

n – Total number of Objects.

There are two different algorithms presented in this paper to deal with horizontally and vertically partitioned data. The basic idea is to count the distance (Euclidean) between two objects and then count the number that exceeds distance threshold. Since, the objects are distributed between different parties the intermediate calculations are held split randomly with participating parties and only the final results released .

2.1.1 Horizontally partitioned data

When the data is horizontally partitioned each participating party collects information about m attributes $A1....Am$ such that various parties collect information about various entities. An example of horizontally partitioned data could be a bank collecting information about credit/debit card transactions for their customers. Here, the attributes (m) are the customers and the information collected is their transactions.

Now, each party will be involved distance calculation. For every object i, its distance from every other j objects will be calculated. If they are both the objects are in the different parties then there exists a distance calculation
protocol between the parties and get random shares of the distance.Another protocol the distance is not shared but binary values „0", „1" are shared such that the party returns 0 if the distance exceeds threshold else it returns 1. Privacy is achieved in this because each parties share their own results such that $rq + rs = 1$ or 0. (Mod F) where, F indicates count of entities. This indicates to get information about the outlier both the shares are required. Nothing can be interpreted from a single share.

Once all distances are calculated, the corresponding zeros or ones are added and thus, the total shares add to 1 or 0 if the distance exceeds the predefined threshold or otherwise. Further, calculating the total share (mod

F) gives us count of entities exceeding threshold or otherwise. The calculation of this sum is a continuous process. Each party computes its share and then passes it to the designated party to calculate its sum and so the total sum is cumulative. Furthermore, the parties exchange in a secure protocol which releases its results only if the sum exceeds the threshold (*p %*).
Let us discuss how to calculate the distance between the two points (X, Y):
Consider two parties (P1, P2) participating in this computation and such that point X,Y is input to P1 and P2 respectively and r1, r2 are the responses. Our aim is to calculate *Distance(X, Y)*. However, for convenience we will calculate square of this distance i.e.:

$$\begin{aligned} Distance^2(X,Y) &= \sum_{r=1}^{m}(x_r - y_r)^2 \\ &= x_1^2 - 2x_1y_1 + y_1^2 + \ldots \\ &\quad \ldots + x_m^2 - 2x_my_m + y_m^2 \\ &= \sum_{r=1}^{m} x_r^2 + \sum_{r=1}^{m} y_r^2 - \sum_{r=1}^{m} 2x_ry_r \end{aligned} \quad \ldots (A)$$

Here, X and Y are represented by values *xr* and *yr* respectively. The first two terms are calculated by P1, P2 respectively and for the third term P1, P2 sets up a secure scalar product protocol.

The algorithm presented in the paper is shown below:

Algorithm 2 Finding DB(p,D)-outliers

Require: k parties, $P_0, \ldots, P_{k-1}$; each holding a subset of the attributes for all objects O.
Require: dt_r : local distance threshold for P_r (e.g., $dt^2 + m_r/m$).
Require: Fields D larger than twice the maximum distance value (e.g., for Euclidean this is actually $Distance^2$), F larger than $|O|$

for all objects $o_i \in O$ **do**
  $m'_0 \leftarrow m'_{k-1} \leftarrow 0 \pmod F$
  for all objects $o_j \in O, o_j \neq o_i$ **do**
    P_0: Randomly choose a number x from a uniform distribution over the field D; $x' \leftarrow x$
    for $r \leftarrow 0, \ldots, k-2$ **do**
      At P_r: $x' \leftarrow x' + Distance_r(o_i, o_j) - dt_r \pmod D$ {$Distance_r$ is local distance at P_r}
      P_r sends x' to P_{r+1}
    end for
    At P_{k-1}: $x' \leftarrow x' + Distance_{k-1}(o_i, o_j) - dt_{k-1} \pmod D$
    {Using the secure comparison protocol (Section 3.3)}
    $P_0 \leftarrow m_0$ and $P_{k-1} \leftarrow m_{k-1}$ such that:
    if $0 < x' + (-x) \pmod D < |D|/2$ **then**
      $m_0 + m_{k-1} = 1 \pmod F$
    else
      $m_0 + m_{k-1} = 0 \pmod F$
    end if
    At P_0: $m'_0 \leftarrow m'_0 + m_0 \pmod F$
    At P_{k-1}: $m'_{k-1} \leftarrow m'_{k-1} + m_{k-1} \pmod F$
  end for
  {Using the secure comparison of Section 3.3}
  $P_0 \leftarrow temp_0$ and $P_{k-1} \leftarrow temp_{k-1}$ such that:
  if $m'_0 + m'_{k-1} \pmod F > |O| * p\%$ **then**
    $temp_0 + temp_{k-1} \leftarrow 1$ {o_i is an outlier}
  else
    $temp_0 + temp_{k-1} \leftarrow 0$
  end if
  P_0 and P_{k-1} send $temp_0$ and $temp_{k-1}$ to the party authorized to learn the result; if $temp_0 + temp_1 = 1$ then o_i is an outlier.
end for

This algorithm has a major drawback that the computational complexity is quadratic. Furthermore, the secure protocol used for scalar products make the computation difficult to be realized practically. But with some tradeoff between privacy and cost of computation this can be achieved. Thus, for vertically partitioned data, we have tried to describe another approach.

2.1.2 Vertically Partitioned Data

For maintaining privacy while detecting an outlier in the data which is partitioned vertically, we refer to a publication by scholars of „University of Science and Technology of China‟ named „*Privacy Preserving Outlier Detection over Vertically Partitioned Data*‟ [9].

In this paper two PPOD algorithms are described. Both the algorithms are distance based algorithms which are over vertically partitioned data where no private information is disclosed to any participants. They have used secure multiparty computation which they believe guarantees privacy and security.

There are some shortcomings of distance based outlier detection algorithms as described in [7] such as, the user have to specify the distance parameter which is very difficult to determine and not ranking the outliers. In this paper they have used distance based PPOD algorithm but have included the work from [7] to avoid the shortcomings of distance based algorithms.

They have used a semi-honest model for performing all the operations, i.e. all the participants are semi honest. Semi honest participants mean that all the participants are honest but are curious during the execution. In this paper they have used Paillier cryptosystem [8] which is semantically secure homomorphic system. In homomorphic cryptosystem, sum of two plain text can be obtained by multiplying two cipher texts and decrypting the result of the multiplication.

Definition of Outlier as per the paper: Given a k and m, a point p is an outlier if no more than m-1 other points in the data set have a higher value for Dk than p where Dk represents the distance of the kth nearest neighbor of a point. This means that computing all points‟ Dk firstly, then we rank all points on the basis of their distances to their kth nearest neighbor and the top m points with the maximum Dk values in this ranking are considered outliers. [7]

As per the paper when considering vertically partitioned data, each party has information of subset of global attributes for common entities wherein the union of all the attribute subsets from all parties equals the global attributes. Moreover, the attribute subsets are disjoint.

Algorithm 1: For two parties is secured

Security for Alice:

Bob has Alice‟s encrypted data but he does not have Alice‟s private key therefore Bob cannot decrypt it and thus privacy of Alice‟s data which is sent to Bob is maintained. Bob has final true position ids of outliers but he cannot infer any ranking information about this outliers.

Security for Bob:

This algorithm adds random values to all the data bob sends. The data Bob sends to Alice is permuted with random permutation which Alice has no idea about and only bob knows that random permutation. In addition the distance vector which is sent to Alice is permutated by another random permutation and similar to first random permutation only bob knows about it.

Alice selects the Kth nearest neighbor and distance from each element in the data without knowing any information as it is permutated with a random variable. After having distance from all the elements in the vector Alice selects top m-position ids for outlier without knowing the original points of outliers as they are permutated.

Algorithm 2: For Multi parties is secured

All the parties (P1,P2…..Pt-1) involved generate a public key *Kp* from a uniform random distribution. Pt generates *Kp* so that the encryption is perfect or sure. This key is used to encrypt the distance vector. All the parties involved send their encrypted distance vector to first party. As P1 knows the dimensionality of the distance vector he simulates it by a uniform distribution. Pt then encrypts the decrypted distance vector and sends position ids to P1. Since P1 knows the random permutation he generates position ids as he receives it from Pt. This generated position ids are sent to all the parties and all the parties can simulate it as this is the final result.

3.2 Density Based Approach

Density based approach of outlier detection is used when it is more interesting in finding rare instances of an outlier rather than finding patterns. Such requirements exist in applications like criminal activity detection. Thus, we are going to first explain the process of detecting an outlier and then we will talk about how to achieve privacy. For this purpose we will refer to a publication named *'LOF: Identifying Density-Based Local Outliers'*. [2]

According to existing work in papers shows that outlier as binary property i.e. an object in a database is an outlier or not. But this paper believes that sometimes it is meaningful to assign a degree to an object of being an outlier. The degree is called local outlier factor. It depends on how isolated an object is from its neighbors. They believe that LOF can be used to find the outliers which are meaningful but they cannot be found out from existing methods.

Outlier detection is largely based on clustering algorithms. And according to clustering, outliers are object that lies outside the clusters. According to clustering outliers are objects having binary property.

According to the paper the LOP is based on a single parameter of MinPts, which is the number of nearest neighbor used to define local neighborhood of an object. In this paper they will show how this affects LOF and how to choose MinPts properly.

Definition (Hawkins-Outlier):

It states Outlier as an observation that deviates so much from normal observation that it creates suspicion as if it is generated from different mechanism.

In this paper they have used o, p, q to denote objects in the dataset. Distance between p and q is denoted by d(p,q). C is used to denote a set of objects with the intuition that C sometimes becomes a cluster. Minimum distance between p and q which belong to C is denoted by d(p,C).

Outliers are object that are situated far from its local neighborhood with respect to density are called "Local Outliers".

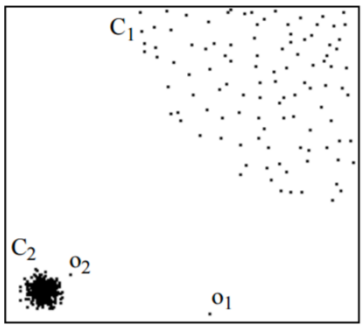

Fig (2.1) – 2-d dataset DSI

In the fig(above there are 502 points, 400 objects in cluster C1, 100 points in cluster C2 and two additional point o1 and o2. According to Hawkins both o1 and o2 are outliers. But according to standard distance based outlier detection algorithm only o1 is an outlier. This is because for a point q in C1 the distance between its neighbors is more than the distance between o2 and C2.

This is the scenario when the global view is not satisfactory when clusters of different densities exist.

Local Outlier Factor of an object P
The Local outlier factor of an object P is defined as

$$LOF_{MinPts}(p) = \frac{\sum_{o \in N_{MinPts}(p)} \frac{lrd_{MinPts}(o)}{lrd_{MinPts}(p)}}{|N_{MinPts}(p)|}$$

From the above equation we get the LOF of object p that is the degree to which we call an object, outlier. It is the average of the ratio of local reachability density of p and those of P‟s MinPts-nearest neighbor. We can see from the equation that if the p‟s local reachability density is less and p‟s local reachability of MinPts - nearest neighbor is high the LOF of p is high.

3.2.1 Privacy-preserving LOF outlier detection

It is important to design a Privacy preserving outlier detection algorithm as the data on which LOF will run is split among various participants who do not want any leakage of their sensitive information. We will discuss an approach from publication "*Privacy-preserving LOF outlier detection"* [10]. This paper proposes a protocol for privacy-preserving LOF Outlier Detection.

Privacy-preserving distance based outlier detection has two important limitations. First is, the distance parameter which is very important in outlier detection is very hard to determine. And second is their applications are very limited. They are inefficient when there are many clusters. To overcome this limitations privacy preserving density based outlier detection is designed.

The main challenge of computing LOF (p) securely that all intermediate values require to generate LOF (p) can leak information and thus should be hidden from the parties involved. For example the parties involved may deduce which objects are more similar to object p and some objects are located in the small region based on Nkp and k-distance (p).

This paper proposes an additive secret sharing of intermediate values to maintain the privacy. For example the value of x is split into two random variable such as x=x1+x2.

Building Blocks required for building a privacy preserving algorithm:

Additive secret sharing ([[x]]): x1+x2 = x(mod q) where x1 and x2 are chosen from uniform distribution.

Homomorphic encryption: A cryptosystem is additively homomorphic if D(E(m1)*E(m2)) = m1+m2.

Permutation Protocol: The input to this protocol is string of x i.e. [[x1]….[[xn]] and output is [[x (1)]]…[[x (n)]] where is chosen by one party and not known to other parties. is a random permutation.

Secure Scalar Product Protocol: In this paper they have implemented share shuffling protocol by using permutation twice. Here the value of is not known to any of the parties involved. For example A chooses a vector and B chooses another vector. After Permutation A gets c1 and B gets c2 such that c1+c2 is sum of the product of all the elements in the two vectors.

3.2.2 Privacy-Preserving LOF Outlier Detection Protocol:

Protocol 1 Privacy-Preserving LOF Outlier Detection Protocol

Input: $[\![D]\!]$, parameter k

Output: $lof(o_i), (i = 1, 2, \ldots, n)$

Step 1: Initialization and query k-NN.

Step 2: For each object, calculate the square of reachability distances of it w.r.t. the objects in its k-NN set.

Step 3: Transform the square of distances to distances, and calculate *local reachability density* values for all the objects.

Step 4: For each object, obtain *local reachability density* values corresponding to the objects in its k-NN set, and calculate the final output.

Suppose 2 parties are involved A and B. Both A and B computes distance (square) independently for the two objects. They share a distance matrix. The input to secure K-NN Query is additive shares of S and the output is share of k elements such that the first k smallest elements in S.

Protocol 2 Secure k-NN Query

Input: $S = (\llbracket s_1 \rrbracket, \ldots, \llbracket s_n \rrbracket)$
Output: $T = (\llbracket t_1 \rrbracket, \ldots, \llbracket t_k \rrbracket)$
Step 1: $\mathcal{A}$ and $\mathcal{B}$ run sharing shuffle protocol to obtain randomly reordered shares $\llbracket c_1 \rrbracket, \ldots, \llbracket c_n \rrbracket$ of $s_1, \ldots, s_n$.
Step 2: Parties choose $l = \lceil log_2 n \rceil$ and then locally compute shares of $2^l * c_1 + 1, \ldots, 2^l * c_n + n$.
Step 3: After that $\mathcal{A}$ and $\mathcal{B}$ run the k-NN algorithm to find k-first elements.

This protocol act as a subroSutine in privacy-preserving outlier detection protocol. The output of K-NN query should not leak or reveal any information that cannot be found out from output of outlier detection and input of participants. Most K-NN Query outputs information and thus they cannot be used in LOF-Outlier detection. The solution to this problem is using a share shuffle protocol. This will not reveal any information but will tell us what element in new distance sequence has larger values.
In this protocol the input sequence is additively split into two-party and it has been permuted. Thus, the secure select can be used as an intermediate process. Thus both the parties will know which element has smaller values in the new sequence but none of them knows that which element corresponds to which element in the sequence.

Based on the K-NN query protocol both the parties involved will get the share of k-distance of each point, the garbled K-NN set information and their own permutation key for each point.

Securely calculating LOF values Algorithm as per the paper

Protocol 4 PPLOF-Step 2

Input: $[\pi_i], \llbracket kd_i \rrbracket, \llbracket T_i \rrbracket, pos_i, i = 1, \ldots, n$
Output: $\llbracket square\_rdis_{i1} \rrbracket, \ldots, \llbracket square\_rdis_{ik} \rrbracket, i = 1, \ldots, n$
for $i = 1$ to n **do**
Parties use the following two items as input:
1. $\llbracket kd_1 \rrbracket, \ldots, \llbracket kd_{i-1} \rrbracket, \llbracket kd_{i+1} \rrbracket, \ldots, \llbracket kd_n \rrbracket$ (input sequence).
2. $[\pi_i]$ (input shuffle key).

They use $[\pi_i]$ to shuffle the input sequence.
According to the positions pos_i, they obtain the shares of desired k-distance (square) values.
Two parties run SMP k times to obtain shares of desired reachability distances.
end for

Protocol 5 PPLOF-Step 3

Input: $\llbracket square\_rdis_{i1} \rrbracket, \ldots, \llbracket square\_rdis_{ik} \rrbracket, i = 1, \ldots, n$
Output: $\llbracket lrd_i \rrbracket, i = 1, \ldots, n$
for $i = 1$ to n **do**
for $j = 1$ to k **do**
$\mathcal{A}$ and $\mathcal{B}$ calculate $\llbracket 4^\lambda square\_rdis_{ij} \rrbracket$.
Then they engage in SSRP on $\llbracket 4^\lambda square\_rdis_{ij} \rrbracket$ to obtain shares of $\llbracket rdis_{ij} \rrbracket$.
end for
end for
for $i = 1$ to n **do**
Two parties calculate $\llbracket sum\_rdis_i \rrbracket = \llbracket rdis_{i1} + \cdots + rdis_{ik} \rrbracket$.
The parties run SDP to share $\lfloor 4^\lambda k\delta / \llbracket sum\_rdis_i \rrbracket \rfloor$.
end for

The entire process of PP LOF outlier detection can be explained with the help of a diagram:

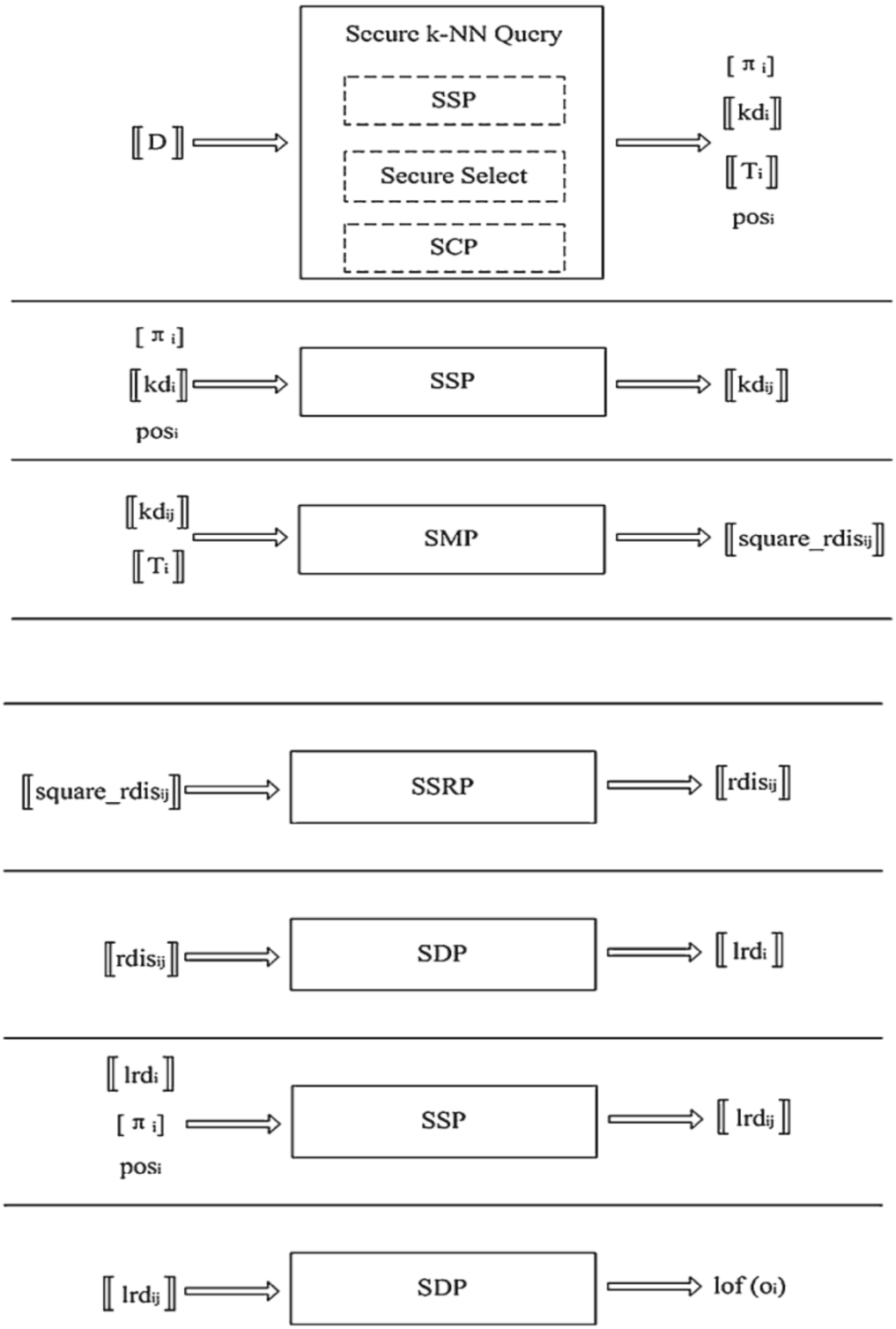

Fig (2.2) – *The process of privacy preserving calculating LOF value.*

Protocol 6 PPLOF-Step 4

Input: $[\![lrd_i]\!], [\pi_i], pos_i, i = 1, \ldots, n$
Output: $lof(o_i)$
for $i = 1$ to n **do**
The parties takes the following two items as input:
1. $[\![lrd_1]\!], \ldots, [\![lrd_{i-1}]\!], [\![lrd_{i+1}]\!], \ldots, [\![lrd_n]\!]$ (input sequence).
2. $[\pi_i]$ (input shuffle key).

They use $[\pi_i]$ to shuffle the input sequence.
According to the positions pos_i, they obtain the shares of desired local reachability density values.
Two parties calculate $[\![sum\_lrd_i]\!] = [\![lrd_{i1} + \ldots + lrd_{ik}]\!]$.
Then they run SDP to obtain $[\![2^{\lambda} sum\_lrd_i]\!] / [\![lrd_i]\!]$.
end for

This algorithm cannot be applied on horizontally partitioned data but SSP can be applied to get shares of distance between each object. It is hard to design the algorithm when the data is horizontally partitioned between two parties even if they are using the subroutines.

In this paper [10] they have implemented a secure privacy preserving outlier detection using LOF which works for vertically partitioned data. They have designed a K-NN query protocol which is secure. They have used same permutation as used in in K-NN query protocol to map values to object so that they are found easily.

4. Conclusion (10pt)

In this report we have summarized two major approaches of finding or detecting and outlier which are distance/proximity based approach and density based approach. We then discussed few works of scholars in the field of data privacy and their attempts to keep the data differentially private while detecting an outlier. We analyzed (at the end of each section) their publications and tried to mention merits as well as demerits of implementing their proposed algorithms in real life.
Currently, efforts are being made to realize these algorithms using software packages. A further study of privacy protection in other approaches of outlier detection such as depth based approach can be used to create a more extensive report.

References(10pt)

[1] J. Vaidya, C. Chris, "Privacy-preserving outlier detection", Proc. 4th IEEE International Conference on Data Mining, Brighton, UK, 2004, 233-240, IEEE Computer Society Press.
[2] M. Breunig, H. Kriegel, R. Ng, J. Sander, "LOF: identifying density-based local outliers", Proc. the 2000 ACM SIGMOD international conference on Management of data, Dallas, Texas, USA, 2000, 93-104, ACM Press.
[3] F. Zhang, H. Chang, "A privacy-preserving outlier detection protocol based on probabilistic public-key encryption", Journal of Computer Research and Development,2006, 43(Suppl.):270 - 274.
[4] A. Xue, X. Duan, H. Ma, W. Chen, S. Ju, "Privacy preserving spatial outlier detection", Proc. 2008 the 9th International Conference for Yong Computer Scientists, Hunan, China, 2008, 714-719, IEEE Computer Society Press.
[5] Y. Huang, Z. Lu, H. Hu, R. Li, "Privacy preserving outlier detection", Acta Electronica Sinica, 2006, 34(5):796-799.
[6] Goldreich. The Foundations of Cryptography, volume 2, chapter General Cryptographic Protocol, Camridge University Press, 2004.
[7] Sridhar Ramaswamy, Rajeev Rastogi and Kyuseok Shim, "Efficient algorithms for mining outliers from large data sets", In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, 2000, pp.427-438.
[8] Pascal Paillier. "Public-key Cryptosystems Based on Composite Degree Residuosity Classes", In Proceedings of Advances in Cryptology- EUROCRYPT"99, vol.1592 of Lecture Notes in Computer Science. Springer-Verlag, 1999, pp.223-238.
[9] Zhengyou Zhou, Liusheng Huang, Yang Wei, Ye Yun NHPCC, Depart. of CS. & Tech. University of Science and Technology of China Hefei, China
[10] Privacy-preserving LOF outlier detection by Lu Li, Liusheng Huang, Wei Yang, Xiaohui Yao, An Liu