Available online at: http://www.basra-science-journal.ogr

ISSN -1817 -2695

Received 18-1-2016, Accepted 22-11-2017

Secure mining of the cloud encrypted database

Saba Abdul W. Saddam
Computer science, College of Computer Science & Information Technology
University of Basrah

Abstract

Due to the stunning characteristics of cloud computing, such as tremendous scalability, elasticity, cost-efficiency, pay-as-you go, and storage solutions, many enterprises and individuals are motivated to outsource their data to cloud service providers for availing its benefits. Protecting and preserving the privacy of these data represent a persistent barrier from adopting the cloud computing. Mining the cloud data may be misused for a variety of purposes. To counter this problem, we propose a secure framework for mining the cloud data in a privacy preserving manner. Secure ***k Nearest Neighbor*** **(kNN)** classifier is used in this paper.

In this work, we preserve all the restrictions that we specify privacy and success to exclude the third party from the mining process. We test our secure classifier with different parameters to explain its influence on the accuracy and privacy the suggested classifier.

Keywords : Cloud, Privacy, Encryption, Data mining.

1. Introduction

Many good features make cloud computing, the most attractive technology in these days. Cost-effective, pay-as you go, great scalability and flexibility are more attractive features that make many companies, government organizations and even individuals to leveraging the significant benefits of this new technology as a mean to store and process its private data. Despite these features, there exist some undesirable demerits in cloud computing that preclude many potential enterprises from reaping and utilizing the benefits of this modern paradigm. The main unwanted feature is the loss of control which is a natural consequence from the fact that cloud services process users' data in machines that the user doesn't own and even doesn't know. Another unfavorable feature is the changed location of the stored data where cloud service providers may exchange users' sensitive and personal data and may transfer it to a compromised location. The above mentioned drawbacks arises many questions and concerns about the security and privacy of the users' personal information.

As stated in [1] user's fears about their data privacy threats can be classified into two scenarios: the first one is the theft and loss the personal data from the cloud service provider. For treating this threat, many mechanisms must be applied to control access, modify, copy, disclosure of the personal information. This direction is out the scope of our work. The other scenario is the mining of multiple databases belonging to different data owners. The concern here comes from leaking additional personal information as a result from the mining process.

The huge computing power and the abundant date available in the cloud make the work of the miner/attacker easy and therefore increase the data owners' worries about preserving their private data privacy [2].

In this paper, we intend to alleviate users' concerns through designing an approach in order to mine the cloud data in an effective, accurate way and at the same time preserve the privacy of user's data. Many business companies -even competitive ones- with sensitive sales data may wish participate and share their private data to know the aggregate trends without leaking the trends of their individual stores. The crucial question is how these companies can mine their private databases and get accurate results (utility) without revealing any private data beyond the final results (privacy). The key to tackle this dilemma is via using the ***Privacy Preserving Data Mining algorithms*** (**PPDM**).

PPDM algorithms aim at providing a trade-off between sharing information for data mining analysis, on the one side, and protecting information to preserve the privacy of the involved parties on the other side. PPDM algorithms can be classified according to the number of parties on which mining will be applied into two main types:

Single party PPDM: Many techniques have been suggested here like k-anonymity and randomization. The K-anonymity technique uses generalization and suppression methods to prevent some pseudo-identifiers (such as birth date, zip-code) from conducting with another public records in order to identify uniquely the private records. This technique tries reducing the granularity of representation of the data in such a way that a given record cannot be distinguished from at least (k − 1) other records. For further details see

[3, 4, 5, 6]. Two strategies are commonly used to merge the k-anonymity and data mining task: Anonymize-and-Mine, where we first anonymize the private data and perform mining over the result, nevertheless Mine-and-Anonymize, perform mining on the private data then anonymize the result. In the randomization method [7, 8], the pretreated data is gained by adding or multiplying some noise to the original data. The mining operations conduct on the aggregated distribution of the results. The main drawback of this method is that the utility of data is stained as a result of the added noise.

Distributed Parties PPDM: These algorithms use secure and cryptographic protocols for the purpose of maintaining the privacy during mining these parties. These protocols add an additional burden to the analysis task. The data of these multi parties may be distributed in two ways:

Horizontal Partitioning: In this case, the different parties may have different sets of records containing the same attributes.

Vertical Partitioning: In this case, the different parties may have different attributes of the same sets of records. It is worth noting that the privacy notation here is no party should learn any new information beyond the final results of the mining process. Distributed PPDM algorithms are similar closely to the ***Secure Multiparty Computing (SMC)***, a field of cryptography originated from Yao's Millionaires' enigma [9]. SMC deals with the cooperation of k parties to compute the global function f(x_1, x_2, ..,x_k) in a secure and private manner without neither disclosing their private data to the other parties nor using any ***Trusted Third Party (TTP)***.

In our work, we intend to classify securely the single database in the cloud while maintaining the privacy of the private data. We deem that the miner has a special database and at every time she picks up one record (query) to classify it according to all the partitioned data in the cloud database. We assume that both cloud data –each party data- and miner data are all encrypted by their owners for security and privacy objectives. After submission the query to the cloud, the single party participates in a secure computing without the intervention of the TTP to calculate the class of that query and return the result to the miner. Practically, we use the KNN classifier. Our framework fulfills, from one hand, all the privacy requirements and from another hand maintains the trade off between the accuracy and the efficiency.

The remainder of this paper is organized as follows. In section 2, we discuss the related works on both the privacy preserving data mining algorithms and the new directions related to the privacy of cloud computing . Section 3 describes the framework motivation, such as the definition problem and specifies our requirements, followed by section 4, which gives a brief explanation to some related primitives. Our framework is illustrated in section 5, which presents in further details the proposed scenario and the privacy preserving suggested classifiers. Section 6, present the experimental result. We conclude the paper in section 7.

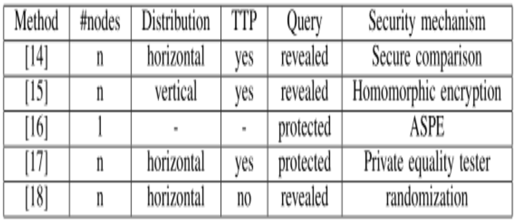
2. Related Works

Recently, there is a growing emergence of privacy preserving data mining algorithms to mitigate the raised side effects of mining private data. The new patterns and trends gathered and integrated from the mining operations have increased chance to break the privacy of the raw data.

The work in [10] was the pioneer for building securely an ID3 decision tree over horizontally distributed data into only two parties. The authors introduced many primitive secure sub protocols such as secure log algorithm, secure polynomial evaluation. Another work [11] has been presented to get the association rules among horizontally partitioned data. The effort of [12] was dedicated to perform the Naive Bayes classifier on horizontally distributed data. The authors of [13] suggested a solution maintaining the privacy in SVM classifier. The earlier work in secure KNN classifier was presented in [14] on horizontally partitioned data. The researchers used the secure compression to compare the k-nearest neighbors between the parties. So, no party knows the final result of the comparison. The results from all parties then are integrated and permuted and delivered to the TTP to complete the classification task. The homomorphic encryption scheme is the key concept for securing the KNN classifier on vertically partitioned data in [15]. The above two methods work under the assumption of revealing the instance query that wants to be classified. The work in [16] maintains this violation of privacy via even encrypt the instance query value in addition to all the training values. The trend of [16] was to perform k-nearest neighbor (kNN) computation on an encrypted single database. The researchers developed a new asymmetric scalar-product-preserving encryption (ASPE), which will be explained later in further details. The work presented in [17] is highly related to our work. The effort was to design a secure KNN classifier over encrypted multiple databases in the cloud. Many third parties have been used in this work such as the equality tester, who will measure the similarity between the encrypted values of all the training nodes and the test instance by using Jaccard similarity function. The other third party is the coordinator, who receives the k-local lists from all the nodes in the form (class no., similarity score) and merge the received lists into global list and send it to the classifier to complete the work. Despite the efficiency of this work, its main drawback is the frequent depending on the TTPs and the disclosing of much additional information to the classifier represented by the global list. These drawbacks will be overcome in our suggested work. The authors of [18] made a trade off between the three important requirements of the secure kNN classifier, namely efficiency, privacy, and accuracy. They succeed in performing private classification on multiple parties without using the TTP. They used the probabilistic randomization and secure addition techniques to enable the participating parties to perform the classification in a private manner. The weakness of their work is the revealing of the instance test value which is the drawback that we try to fix in our framework.

Apart from the PPDM, in the recent years, a new research direction pertinent to the privacy in the cloud computing has been emerged. In fact, this direction addresses the problem of maintaining the privacy of the keyword search over the encrypted documents in the cloud. Sometimes, it is utmost important to prevent the cloud service provider from knowing the keyword, access pattern, search pattern, and quest the content of the stored documents. In this field, [19] showed how can we retrieve securely the documents from the cloud provider which contain the private multi key words in a ranked fashion, and in the same time restricted with the above mentioned privacy requirements. In [20] another research direction related to the privacy of the cloud has been applied. It is try simply to measure -in a secure way- the similarity between the data owner DO access control policies and their corresponding in the cloud service providers SPs.

TABLE I: Characteristics of the secure KNN classifier.



Method	#nodes	Distribution	TTP	Query	Security mechanism
[14]	n	horizontal	yes	revealed	Secure comparison
[15]	n	vertical	yes	revealed	Homomorphic encryption
[16]	1	-	-	protected	ASPE
[17]	n	horizontal	yes	protected	Private equality tester
[18]	n	horizontal	no	revealed	randomization

3. Framework Motivations

In this section, we describe the problem on which we concentrate in this paper and explain the main requirements which we seek to fulfill in our framework.

3.1 Problem Definition

We assume the existence of only a single party contains private database. Such database contains a set of records (instances) represented as a vector of *d* attributes, and belongs to a single class. Due to the security concerns in the cloud environment, database is encrypted before shipping it to the hostile medium of the cloud. There is also another database belongs to the miner who picks up –at every time- a single instance query and wants to classify it among the single database. One of the most important contributions in this work is that we also assume the encryption of even the miner database; in other words, the instance query values are also encrypted. The enforcement of the privacy restrictions is maintained throughout all the phases of classification task. The parties disclose as little as possible private information during both the training phase –building the classifier- and the testing phase. In the next section, we will elaborate the speech on the privacy restrictions that has been adopted and maintained in our work.

Moreover, we broke successfully the traditional prevailing use of the trusted third party to manage the classification process among multiple parties. We observed that it is very difficult to find a TTP which has the trust of all the parties. So, we exclude the TTP from the classification work and limit its role to only generate a pair of keys and purplish them to the miner and the owners of the databases in the cloud to encrypt their private data. Our excluding of TTP form the classification job is considered a pivotal advance towards achieving more private and secure environment to conduct the mining in the cloud.

3.2 Framework Requirements

The most important three objectives to design any secure classification algorithm are its *utility, efficiency*, and *privacy*. The utility of the privacy preserving classifier is its ability to predicate the right results for the new instances. The algorithm efficiency describes in this case its ability to deal with huge databases. While the privacy condition meets the privacy requirements of all the parties participating in the mining process.

Practically, it is very difficult to design a privacy preserving classification algorithm which meets all the above stated objectives. The selected technique in the PPDM algorithm has a high impact on the trade off between these objectives. For example, cryptographic techniques for privacy-preserving distributed data mining do not allow easy trade-off between privacy and accuracy. In contrast, the randomization techniques, allow adjusting the level of privacy while potentially reducing utility. Generally speaking, it is desirable here to maintain maximum utility of the data without compromising the underlying privacy restrictions.

Beside the tradeoff between the above three directions, we try meeting many different types of privacy restrictions. These privacy restrictions encompass the following:

1. Instance Query Privacy: all the previous works in [14, 15, 17, 18] have a common disadvantage represented by exposure the instance query. We succeed to protect the instance query and make the entire mining task over encrypted query.
2. Test Data Privacy: all the private data in the cloud databases are assumed here to be encrypted and never decrypted during the similarity measure. By this restriction we enhance the work in [17] which decrypts the test data to perform the Jaccard similarity measure. The above two privacy restrictions are maintained by using the encryption methods.

4. *Design Primitives*

In this section, we are going to explain briefly some important primitives which have been used during our work.

4.1 *KNN: K-Nearest Neighbor classification problem*

The KNN classifier is a simple and powerful instance based method for classify the instance query -in the form of d attributes- among the training data set. It finds the group of k points –according to a distance function- in the training set that is much close to the test point, and assigns to the query the predominant class among the k-neighbor points. Commonly, the Euclidean distance measure is used to measure the distance between any two points. The basic distance kNN algorithm is presented in Algorithm 1.

Algorithm 1 Basic kNN Algorithm

```
Input: D, the set of training points; the test object, q, which is a vector of
       attribute values, and V the set of classes used to label the points.
Output: $c_q \in V$, the class of q
Foreach point y ∈ D do
    Compute d(q, y), the distance between q and y;
end
Select N ∈ D, the set of k-nearest neighbors training points for q;
```

$$C_q = \arg\max_{v \in V} \sum_{y \in N} I(v = class(p_y)) \quad \ldots(1)$$

Where (I) is an indicator function that returns the value 1 if its argument is true and 0 otherwise.

4.2 *Asymmetric Scalar Product Preserving Encryption (ASPE)*

Cryptography is often described as a perfect treatment to protect the cloud data. The problem is that encryption will cut down data use. Searching, computing and indexing the data become unpractical. State-of-the-art cryptography supports many primitive tools to handle this problem. Recently, versatile encryption schemes have been developed that allow operation and computation on the cipher text. Searchable encryption [21, 22, 23, 24], private information retrieval [25], homomorphic encryption [26], and distance preserving transformations (DPT) [27] are famous examples in this research field.

In DPT approach, secure KNN can work on encrypted data since the distance between any encrypted-transformed- data points E(DB) is the same as that between their corresponding original data points DB. Unfortunately, this nice approach has been proved in [28] to be not secure in practice. Simply, if the attacker gets the encrypted database E(DB) and some of its original points, then it is very easy for him to recover the whole DB.

Wong et al in [16] developed a secure KNN search over encrypted points of d-attributes. Their work gets over the weakness of the DPT by an encryption function doesn't disclose the distance information. They started from the fact that KNN search doesn't need for computing the exact distance, just the distance comparison is necessary. Given points p_1, p_2 and the query q, there are three different scalar products are defined by [16]:

- type-1: self-product, i.e., the product between a point and itself (e.g. $||p_1||^2$ or $||p_2||^2$);
- type-2: point-to-query product (e.g. $p_1 \cdot q$ or $p_2 \cdot q$);
- type-3:point-to-point product (e.g. $p_1 \cdot p_2$).

The suggested ASPE scheme preserves only the second type without either type-1 or type-3. They proposed two distinct encryption functions $E_Q(\ )$ and $E_T(\ )$ to encrypt the query point , data point, respectively. ASPE uses the matrix M (d+1) *(d+1) and its inverse M^{-1} as an encryption key. Each data point - in the form of d-attribute vector- is encrypted as:

$$P' = E_T(p, M) = M^T p$$

Where, M^T is the transforming form of M. In the same way the query point q encrypted as:

$$q' = E_Q(q, M^{-1}) = M^{-1} q$$

For more protection, the exact value of the norms ||p|| is also protected by expanding the original points p,q to become as the following forms:

$$P'' = (p^T, -0.5||p||^2)^T$$
$$q'' = r(q^T, 1)^T$$

Where, *r* is a random variable greater then 0. *p'* and *q'* depend now on the new values of p" and q". According to ASPE scheme, we use the following theorem (from [16]) to determine which one of p_1 or p_2 is nearest to q.

Theorem 1.

$$(p'_2 - p'_1).q' > 0 \text{ Iff } d(p_2,q) > d(p_1,q)$$

Proof.

$$(p'_1 - p'_2).q' = (p''_1 - p''_2)^T.q''$$
$$= (p_1^T - p_2^T, 0.5||p_2||^2 - 0.5||p_1||^2).r(q,1)^T$$
$$= (p_1 - p_2)^T rq + (0.5||p_2||^2 - 0.5||p_1||^2)r$$
$$= 0.5r\,\{2(p_1 - p_2)^T q - ||p_1||^2 + ||p_2||^2\}$$
$$= 0.5r\{d(p_1,q) - d(p_2,q)\}$$

If $0.5r\{d(p_2,q)-d(p_1,q)\} > 0$ IFF $d(p_2,q)>d(p_1,q)$

5. *Single Party Scenario*

Under the assumption of existing only one party with a single private database in the cloud, we construct our scenario. During the mining operation, the miner selects one instance from his own encrypted test database E(DBM) and desires to return its class from the protected database E(DB) in the cloud in a secure behavior. The framework of this scenario is illustrated in figure 1. TTP is used only to generate the encryption keys for both the miner and the data owner.

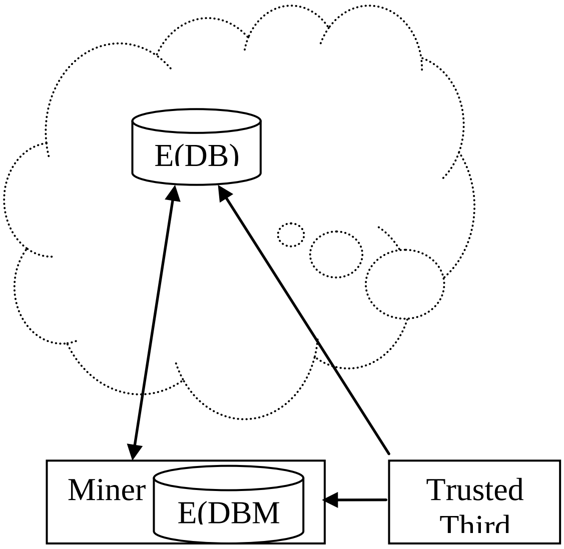


Figure (1): Single database cloud mining.

5.1 *Secure KNN over Single Database cloud*

The famous KNN classifier is constructed to deal with plain text values. We try to modify it to work over encrypted data to fit our privacy requirements. The job of the proposed KNN classifier is divided into two steps. While the first step is corresponded to retrieve the k-nearest points, the second one calculates the query class from these selected points.

In the first step, we make a simple but effective development on the ASPE scheme. As we explained in the previous section, the ASPE decides weather p_1 or p_2 is closer to the query q by computing $(p'_1-p'_2).q'$. Where p'_1, p'_2, and q' are the transformed (encrypted) forms of p_1, p_2, and q, respectively. The matrix M (the key) and its inverse M^{-1} are generated by TTP and given to the data owner and the miner, respectively. The question here is how one can use this scheme to select the k-nearest points from the S points in the database. Our suggestion to solve this problem is to construct the local table T to hold the comparisons between query q and all pairs of the local points pi, i=1..S in DB, we store +1 if the outcome of $(p'_2-p'_1).q'$ is greater than 0 otherwise we store -1. Algorithm 2 illustrates building T table.

Algorithm 2 Building T table

```
  Input query q', and S points.
  Output T table
    For all i=1..S, j=1..S
       If i<j then
          Compute $(p'_i-p'_j).q'$
       If  $(p'_i-p'_j).q' > 0$ then
            T(i,j)= +1
       Else T(i,j)=-1
       If i>j then T(i,j)=-T(i,j)
       T(i,i)=+1
  end
```

Having completed the building of T table, ***the second step*** selects the k-nearest points that have the smallest weights in the table, where the value of the corresponding weights is calculated as the sum of the elements for each row in T table. The rest of the classification task is done as in (4.1).

To make it clearer, let us shows it by an example. Assume that there are 4 points in DB. (p_1, p_2, p_3, and p_4) and their distance to q as follows : $d(p_1,q) < d(p_4,q)$, $d(p_2,q) < d(p_3,q)$, k has the value 3, and let the class values of the four points are (1 2 3 2) in same order. The scheme in its first step computes: $(p'_1.p'_2).q'$, $(p'_1-p'_3).q'$, $(p'_1-p'_4).q'$, $(p'_2-p'_3).q'$, $(p'_2-p'_4).q'$, $(p'_3-p'_4).q'$. Then it builds the table T as explained in table 2:

TABLE II: Building table T

	P_1	P_2	P_3	P_4	Weight
P_1	+1	**-1**	**-1**	**-1**	-2
P_2	+1	+1	**-1**	**+1**	2
P_3	+1	+1	+1	**+1**	4
P_4	+1	-1	-1	+1	0

It is worth noting, that only the bolded cells in the table are need computing. The other ceils are deduced from the computed ones. This smart idea reduces the overall computation overhead. The point p_1 has the lowest weight value (-2) so it represents the closest point to the query, followed by p_4, p_2. The class value will be 2, which is the common class in the class set (1 2 2) of the 3-nearest points. Algorithm 3 illustrates the mining task over the encrypted database in a single party.

Algorithm 3 Secure KNN Classifier over Single Party

```
Input: K>0, instance query q, DB the database of size S on which the q will be classified
Output: Cq class of q
  - Encrypt all the S points in DB.
  - Foreach point P in DB
        Extend P: $P'' = (p^T,-0.5||p||^2)$
        Encrypt P'': $P'= E_T (p'', M) = M^T p''$
  - Protect q
        Extend q: $q''=(rq^T ,r)^T$, where r >0 are random values.
        Encrypt q'': $q'=E_Q (q'',M^{-1})=M^{-1}q''$.
  - Classification:
        Build the T Table (as in Algorithm 2).
        Generate Weight vector by summing the elements of each row in T.
        Select the indexes Ind of K smallest values in Weight.
        Build the set N, which represents the classes of the points that has the indexes Ind.
  - Compute the class Cq of q as equation (1).
```

5.2. Discussion

The above presented scheme still suffers from two serious loopholes. The first one is the revealing of the instance query q to the data owner and this case collides with the query privacy restriction. The data owner can simply infers the exact value of q by computing M.q to yield the vector (rq,r) then simply divides the resulted vector on the last element r to produce (q,1) which is the exact value of q vector. To solve the above problem, we must find a way to hide r value to preclude the data owner from disclosing q vector. The second flaw is the additional computation overhead. Assume there are 4 points in the database; we need for 6 comparisons instead of 4. In the following section, we will present new scheme to handle the above two mentioned problems.

5.3. *Full Secure KNN Classifier over Single Party (FSKSP)*

To reduce the computation overhead of the previous explained scheme, we try to extract the exact distance from q to each point, instead of make a comparison between q and a pair of points. So, a little modification for ASPE scheme is required to reflect our new requirements. The size of the extended form of both q and all p points will be (d+2) as the following:

$$p'' = (p^T ,-0.5||p||^2,1)^T,$$

$$q'' = (rq^T, r,-0.5r||q||^2)^T.$$ where r>0 is a random variable.

The encrypted form of P'' and q'' is:

$$P' = E_T(p'',M)= M^Tp''.$$

$$q' = E_Q(q'', M^{-1}) = M^{-1}q''.$$

The exact distance between the p and q points is -2(p'.q') in case r=1. The following theorem explains the proof of this expression.

Theorem 2

$$(p'.q') = -0.5rd(p,q)$$

Proof

$$\begin{aligned}(p'.q') &= (p'')^T.q''\\ &= (p^T, \quad -0.5 \ ||p||^2 \ ,1) \ . \ (rq,r,-0.5r||q||^2)^T\\ &= (rpq -0.5r||p||^2 -0.5r||q||^2)\\ &= -0.5r(-2pq+ ||p||^2+||q||^2)\\ &=-0.5 \ rd(p,q).\end{aligned}$$

To solve the problem of hiding q value, we want to protect r value as we explained earlier. A simple solution to protect r value is by adding a random value (t) to r. Generally speaking, adding the random value t to r will reduce the accuracy and preserve the privacy of q as we will explain in the experimentation results.

The final version of the algorithm that mines the encrypted database in a single party is illustrated in Algorithm 4.

Algorithm 4 FSKSP

Input: K>0, instance query q, DB the database of size S on which q will be classified

Output: Cq class of q

Miner Side

- Extend q: $q'' = (rq^T, r+t, -0.5r||q||2)^T$. where r,t>0 are two random variables. t is added to hide r value.
- Encrypt q: $q' = E_Q(q'', M^{-1}) = M^{-1}q''$.

Data owner Side

- ***Foreach*** p in DB
    - Extend p: $p'' = (p^T, -0.5||p||^2, 1)^T$.
    - Encrypt p: $P' = E_T(p'', M) = M^T p''$.

- Distance Calculation: compute -2(p'.q') as the distance between the two point p and q.

- Classification
- Select the k-nearest points (N) which have the k-smallest distances from q.
- Calculate the class C_q of q as equation (1).

6. Experimental Results

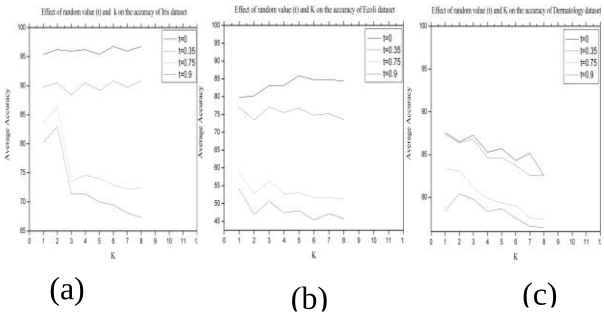
In this section, we make different experiments to assess the accuracy of our classifiers according to the varying of different parameters.

Our experiments depend on three public available datasets. The first dataset, IRIS [30], the best known database to be found in the pattern recognition literature, contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are not linearly separable from each other. Each instance consists of 4 attributes. The second dataset, ECOLI [31], have 336 instances of 8 attributes and classified into 8 classes. It is medical dataset used for predicting the cellular localization sites of proteins in Gram-Negative Bacteria and Eukaryotic Cells. The third dataset, DERMATOLOGY, [32], is also medical dataset used for differential diagnosis of Eryhemato-Squamous diseases which is a real problem in dermatology. They all share the clinical features of erythema and scaling, with very little differences. This dataset contains 366 instances belonging to 6 classes, with each instance having 34 attributes.

During our work, we divide each dataset randomly into two subsets – a training subset which is used for training the classifier- involves 3/4 of the data and test subset involving 1/4 of the data. The obtained results are the average of 10 runs; during each run we randomly partitioned the dataset.

6.1 Effect of Privacy Factor t on FSKSP Classifier Accuracy

In this experiment, we assess the accuracy of the FSKSP by using different values of the random variable *t* to measure its impact on the accuracy of the classifier. We conduct this experiment over different values of *k*. Setting *t* value to 0 illustrates the accuracy of the classifier without privacy restrictions.



(a) (b) (c)

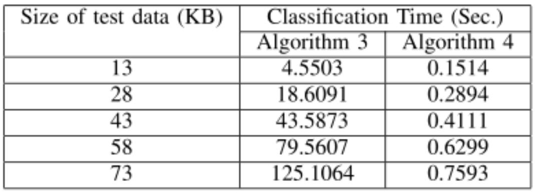
Figure (4): Effect of the privacy factor t on the accuracy of FSKSP classifier

Form figure (4) we notice the great influence of the privacy factor t on the accuracy of the classifier. The higher value of *t* decreases the accuracy and increases the privacy of instance query.

6.2 Efficiency of FSKSP

As we mentioned in single party scenario that the classifier of Algorithm 3 suffers from the low efficiency (speed); accordingly, we have developed Algorithm 4 to enhance the efficiency. In this experiment, we try to compare the speed of the two above algorithms. This experiment ran with k=4, and using the DERMATOLOGY dataset. Table (3) shows the required time to classify different amount of data for the two algorithms. It is easy to see that the first algorithm complete the classification time in a logarithmic fashion, while the second one work in a linear style.

TABLE III: Efficiency of FSKSP.



Size of test data (KB)	Classification Time (Sec.)	
	Algorithm 3	Algorithm 4
13	4.5503	0.1514
28	18.6091	0.2894
43	43.5873	0.4111
58	79.5607	0.6299
73	125.1064	0.7593

7.concluison

In this paper, we have developed a privacy preserving KNN-classifier to mine the encrypted data in the cloud while preserving the all the privacy restrictions. The proposed scheme is applied to classify the data in a single party. The two main contributions of this work are: excluding the third party from the classification task, and protect the privacy of the instance query. Our scheme keeps a balance between the accuracy, privacy, and the computational overhead.

8. References

[**1**] S. Pearson. ***Taking account of privacy when designing cloud computing services***. In *CLOUD'09: Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing*, 2009.

[**2**] R. Chow, P. Golle, M. Jakobsson, R. Masuoka , J. Molina , E. Shi, J. Staddon. ***Controlling Data in the Cloud: Outsourcing Computation without Outsourcing Control***. In *proceedings of the CCSW 2009: The ACM Cloud Computing Security Workshop*, 2009.

[**3**] Pierangela Samarati. ***Protecting respondents' identities in microdata release***. *IEEE Transactions on Knowledge and Data Engineering, 13(6):1010–1027*, November 2001.

[**4**] Valentina Ciriani, Sabrina De Capitani di Vimercati, Sara Foresti, and Pierangela Samarati. k-anonymity. In T. Yu and S. Jajodia, editors, ***Security in Decentralized Data Management***. *Springer, Berlin Heidelberg*, 2007.

[**5**] Roberto J. Bayardo and Rakesh Agrawal. ***Data privacy through optimal k-anonymization***. In *Proc. of the International Conference on Data Engineering (ICDE'05), Tokyo, Japan*, April 2005.

[**6**] Kristen LeFevre, David J. DeWitt, and Raghu Ramakrishnan**.** ***Incognito: efficient full-domain k-anonymity***. In *Proc. of the ACM SIGMOD Conference on Management of Data, Baltimore, Maryland*, June 2005.

[**7**] Agrawal R., Srikant R. ***Privacy-Preserving Data Mining***. *ACM SIGMOD Conference*, 2000.

[**8**] Agrawal D. Aggarwal C. C. On the Design and Quantification of Privacy Preserving Data Mining Algorithms. *ACM PODS Conference*, 2002.

[**9**] Yao, Andrew C. ***How to generate and exchange secrets***. In *Proceedings of the 27th IEEE Symposium on Foundations of Computer Science, pages 162–167. IEEE*, 1986.

[10] Lindell, Yehuda and Pinkas Benny. ***Privacy preserving data mining***. In *Advances in Cryptology – CRYPTO 2000, pages 36–54. Springer-Verlag*, 2000.

[**11**] Kantarcioglu, Murat and Clifton Chris. ***Privacy-preserving distributed mining of association rules on horizontally partitioned data***. *IEEE TKDE, 16(9):1026–1037*, 2004.

[**12**] Kantarcioglu, Murat and Vaidya Jaideep ***Privacy preserving naive bayes classifier for horizontally partitioned data***. In *the Workshop on Privacy Preserving Data Mining held in association with The Third IEEE International Conference on Data Mining, Melbourne, FL*, 2003.

[**13**] Yu, Hwanjo, Jiang, Xiaoqian, and Vaidya Jaideep. ***Privacy preserving svm using nonlinear kernels on horizontally partitioned data***. In *SAC '06: Proceedings of the ACM symposium on Applied computing, pages 603–610, New York, NY, USA. ACM Press*, 2006.

[**14**] Kantarcioglu, Murat and Clifton Chris. ***Privately computing a distributed k-nn classifier***. In Boulicaut, Jean-Franois, Esposito, Floriana, Giannotti, Fosca, and Pedreschi, Dino, editors, PKDD2004: *8th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 279–290, Pisa, Italy, 2004.

[**15**] J. Zhan, L. Chang, and S. Matwin. ***Privacy Preserving K-nearest Neighbor Classification***. *Intl. Journal of Network Security, 1(1):46–51*, July 2005.

[**16**] Wai Kit Wong, David Wai-lok Cheung, Ben Kao, and Nikos Mamoulis. ***Secure knn computation on encrypted databases***. In *Proc. of the 35th SIGMOD international conference on Management of data,*

pages 139–152, New York, NY, USA. ACM,
2009.

[**17**] M. D. Singh, P.R. Krishna, and A. Saxena ***.A Cryptography Based Privacy Preserving Solution to Mine Cloud Data.*** *Proceedings of the Third Annual ACM Bangalore Conference*, 2010.

[**18**] li xiong, s. chitti, and ling liu ***.Mining multiple private databases using a KNN classifier***. *Proceedings of the 2007 ACM symposium on Applied computing.* , 2007.

[**19**] Ning Cao, Cong Wang, Ming Li, Kui Ren, and Wenjing Lou,. ***Privacy-Preserving Multi-keyword Ranked Search over Encrypted Cloud Data***. In *the proceeding of The 30th IEEE International Conference on Computer Communications (IEEE INFOCOM 2011).*

[**20**] Eun-Ae Cho, Gabriel Ghinita, and Elisa Bertino, ***Privacy-Preserving Similarity Measurement for Access Control Policies***. In *Conference on computer communications security, proceedings of the 6th ACM workshop digital identity management*, oct. 2010.

[**21**] Boneh, B., Di Crescenzo, G., Ostrovsky, R., and Persiano, G. ***Public Key Encryption with Keyword Search***. In *EUROCRYPT*, 2004.

[**22**] Song D., Wagner D., and Perrig A. ***Practical Techniques for Searches on Encrypted Data. In IEEE Symposium on Research in Security and Privacy***. 2000.

[**23**] Shen E., Shi E., and Waters B. Predicate Privacy in Encryption Systems. In *TCC*. 2009.

[**24**] Shi E. Bethencourt J., Chan H., Song D., and Perrig A. ***Multi-Dimensional Range Query over Encrypted Data***. In *IEEE Symposium on Security and Privacy.* ,2007.

[**25**] Chor B., Kushilevitz E., Goldreich O., and Sudan M. ***Private Information Retrieval***. *J. ACM, 45*, 6 (1998), 965-981.

[**26**] Gentry C. Fully Homomorphic Encryption Using Ideal Lattices. In *STOC*. 2009.

[**27**] S. R. M. Oliveira and O. R. Zaiane. ***Privacy preserving clustering by data transformation.*** In *SBBD, Manaus, Amazonas, Brazil*, 2003.

[**28**] K. Liu, C. Giannella, and H. Kargupta. ***An attacker's view of distance preserving maps for privacy preserving data mining***. In *PKDD*, 2006.

[**29**] C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M. Zhu. ***Tools for Privacy Preserving Distributed Data Mining***. *ACM SIGKDD Explorations, 4(2),* December 2002.

[**30**] R.A. Fisher In ***ftp://ftp.ics.uci.edu/ml/machine-learning-databases/iris***.

[**31**] Kenta Nakai In ***ftp://ftp.ics.uci.edu/ml/machine-learning-databases/ecoli***.

[**32**] Nilsel Ilter, H. Altay Guvenir In ***ftp://ftp.ics.uci.edu/ml/machine-learning-databases/dermatology.***

التنقيب الأمن لقاعدة بيانات سحابيه مشفرة

صبا عبد الواحد صدام

جامعة ألبصره – كليه علوم الحاسوب وتكنولوجيا المعلومات – قسم علوم الحاسوب

المستخلص

نظرا للخصائص المذهلة لاستعمال الحوسبة السحابية , مثل القابلية الهائلة , المرونة, الكفاءة من حيث الكلفة, الدفع أينما ذهبت و حلول التخزين ,العديد من المؤسسات والإفراد يحفزون للاستعانة بمصادر خارجية لبياناتهم لتغطيه الخدمات السحابية للاستفادة من فوائدها. الحماية والحفاظ على أمنية هذه البيانات يمثل حاجز مستمر لتبني استعمال الحوسبة السحابية.

تنقيب بيانات السحابة قد يساء استخدامه لمجموعه متنوعة من الإغراض ولمواجهة هذه المشكلة اقترحنا في هذا البحث إطار امن لتنقيب بيانات السحابة للحفاظ على سريه ألطريقه واستخدم k الجار الأقرب ***k Nearest Neighbor* (kNN)** كمصنف امن , حافظنا على سريه كل القيود المحددة ,النجاح لاستبعاد الطرف الثالث من عمليه التنقيب واختبرنا المصنف الأمن الذي لدينا مع معلمات مختلفة لتوضيح تأثيرها على دقة وأمنيه المصنف المقترح.

الكلمات المفتاحيه : السحابه , التشفير , الامنيه , تنقيب البيانات .