Secure K-means clustering using homomorphically encrypted dataset

Report submitted in fulfillment of the requirements
for the B.Tech Project

by

Mayank Kumar, Dhawal Jethwani

15074010, 15074006
Computer Science and Engineering, Part- III
Indian Institute of Technology (BHU) Varanasi
Varanasi 221005, India

Under the guidance of

Dr K.K. Shukla

Professor

Department of Computer Science and Engineering
Indian Institute of Technology
Varnasi 221005, India

Dedicated to

Our parents, teachers and friends

Acknowledgments

We would like to express our sincere and profound gratitude towards **Dr K.K. Shukla**, Professor, Deptt. of Computer Science, IIT (BHU), Varanasi for his guidance and constant encouragement throughout the course of this B.Tech. Project.

Place: IIT(BHU) Varanasi

Date: 28th April 2018

Mayank Kumar, Dhawal Jethwani

CSE III Year

IIT (BHU) Varanasi

Certificate

This is to certify that the work contained in this report entitled "**Secure K-means clustering using Homomorphically encrypted data-set**" being submitted by **Mayank Kumar, Dhawal Jethwani** (**Roll No. 15074010, 15074006**), carried out in the Department of Computer Science and Engineering, Indian Institute of Technology (BHU) Varanasi, is a decent work under our supervision.

I wish them all the best in all their future endeavours.

Dr K.K. Shukla
Professor,
IIT(BHU) Varanasi.

Place: IIT(BHU) Varanasi
Date: 28th April 2018

Declaration

We certify that

1. The work contained in this report is original and has been done by us and the general supervision of our supervisor.

2. The work has not been submitted for any project.

3. Whenever we have used materials (data, theoretical analysis, results) from other sources, we have given due credit to them by citing them in the text of the thesis and giving their details in the references.

4. Whenever we have quoted written materials from other sources, we have put them under quotation marks and given due credit to the sources by citing them and giving required details in the references.

Place: IIT(BHU) Varanasi
Date: 28^{th} April 2018

Mayank Kumar, Dhawal Jethwani
IDD Students
Department of Computer Science and Engineering,
Indian Institute of Technology (BHU) Varanasi,
Varanasi, INDIA 221005.

Abstract

Outsourcing data for third party analysis raises data privacy preservation concerns, hence questioning the viability of the system. Data confidentiality is ensured through cryptography techniques. Any form of sophisticated analysis is precluded upon using techniques like Homomorphic encryption, that allow limited amount of data manipulation. We couple encrypted data with additional information for this to be achieved to f facilitate third party analysis. The report is based on a mechanism for Homomorphically secure k-means clustering and the concept of an Updatable Distance Matrix (UDM). The mechanism is fully described and analyzed and the evaluation henceforth shows that the proposed mechanism produces quite similar clustering results as when standard k-means is applied, but in a secure way. Both data privacy and correctness are preserved when the proposed mechanism thus allows the application of clustering algorithms to encrypted data.

Contents

Chapter 1

Introduction

1.1 Overview

Machine Learning and data mining techniques have been used to channelize decisions and improve quality with respect to many application domains. However, the exponential growth in data availability often makes in-house data analysis impractical and expensive. This has fuelled the idea of Data Mining as a Service (DMaaS); third parity analysis using cloud computing facilities [3]. This also opens the door to collaborative data mining where a number of data owners pool their data so as to gain some (for example commercial) mutual advantage. Although DMaaS reduces the cost of managing and analyzing data, and facilitates collaborative data mining, it introduces several challenges, the most significant of which is data privacy preservation. Hence the research domain of Privacy Preserving Data Mining (PPDM) [1, 9, 16]. The typical approach to PPDM is to use some form of data transformation, applied either to the entire data set (for example data perturbation) or selective sensitive data attributes (for example value anonymization) [4]. In perturbation, individual attributes are distorted or randomised by adding noise, whereas data anonymity relies on modifying the data by replacing sensitive values with proxy values or removing the values all together. However, although perturbed data may look quite different from the original data, an adversary may take advantage of properties such as correlations and patterns in the original data to identify the real values [16]. It has also been observed that data cannot be fully anonymized [2]. In addition the anonymization/perturbation of data may adversely effect the quality of the analysis [1, 16].

1.2 Motivation for research work

Data privacy and security can be substantially guaranteed when data is encrypted. However, standard forms of encryption do not support data mining activities, which typically require the comparison of records to determine their similarity (or otherwise). Several encryption schemes have been developed to enforce privacy and confidentiality while at the same time supporting some manipulation of encrypted data. Searchable Encryption (SE) [17] is an example of an encryption scheme that preserve data privacy while supporting different kinds of search operations over encrypted data. However, such search capabilities still do not readily support data mining activities. One potential solution is Homomorphic Encryption (HE), a family of emerging encryption mechanism that supports a limited number of simple mathematical computations. The precise nature of the mathematical capabilities supported by HE schemes is dependent on the nature of the adopted scheme. However, there is no HE scheme that provides all the mathematical operations that we might want so as to perform data mining activities; for example the similarity calculation operations required in the case of data clustering [12, 15]. The solutions that have been proposed to date all entail a significant element of data owner participation so that operations not supported by the selected HE mechanism can be conducted using unencrypted data by the data owner. The degree of data owner participation is such that the only reason for using a third party is to support cooperative data mining. For example in the context of k-means cooperative clustering, as described in [7] and [13], local cluster centroids are shared on each iteration so that global centroids can be calculated on each k-means iteration, the majority of the work is conducted by the data owners. In the case of a single data owner there seems little point in conducting third party k-means clustering if most of the work needs to be conducted by the data owner. The challenge is therefore to limit the amount of data owner participation.

Given the above, this paper presents a k-means clustering mechanism that limits interaction with data owners by using the concept of an Updatable Distance Matrix (UDM). More specifically the idea is that data is stored, using third party storage, in an encrypted form, together with an associated UDM. The data owner can then request the data to be clustered, specifying the number of clusters k, as required. For the clustering to operate correctly the UDM matrix needs to be updated on each iteration of the k-means algorithm. To do this the differences between the centroids of iteration i and i-1 are calculated in encrypted form and returned to the data owner for decryption, after which the decrypted differences (as real numbers) are returned to the third party in the form of a Shift Matrix (SM) which is then used to update the UDM ready for iteration i; this is the only data owner participation that is required. Using this mechanism, as will be made clear in the paper, the amount of data owner participation is significantly reduced (by a factor of n, where n is the number of records in the dataset). Furthermore, although the idea is presented using the well known k-means clustering algorithm [12], it can, with some adjustment, be extend to encompass alternative clustering mechanisms. It should also be noted that the nature of the encryption used is Liu's scheme [10], which supports addition and subtraction of cypher texts, and division and multiplication of cypher texts by real numbers.

1.3 Organisation

The rest of this paper is organised as follows, in Section 2 a review of related work is presented. Section 3 provides the fundamental background concerning standard k-means clustering and Liu's Homomorphic Encryption scheme. Sec- tion 4 explains the UDM concept as proposed in this paper. The next section, Section 5, presents the proposed Secure k-means cluster over the homomorphi- cally encrypted data founded on the UDM concept introduced in the previous section. Method and results of experimenting the proposed algorithm have been discussed in section 6. Finally, section 7 concludes the paper with some ideas concerning future work to improve proposed mechanism.

Chapter 2

Related works

The main challenge of third party (k-means) data privacy preserving clustering using encryption is that, although the HE schemes used support a range ofarithmetic operations that can be applied to cypher-data, there are some operations that are not supported. Mechanisms have been developed to address this challenge. These can be categorised as featuring either Multi-Parity Computation (MPC) or Partial Third-party Computation (PTC). Both feature significant data owner participation.

MPC is only applicable where the desired clustering is to be undertaken with respect to data belonging to two or more data owners. The basic idea is that the majority of the processing is conducted in-house by the data owners, only the centroids are shared on each k-means iteration [6]. Examples can be found in [13] and [7]. In [13] the data owners, on each k-means iterations, generate local clusters using their individual datasets. Each party then calculates the centroid of their local cluster, encrypts this and shares the encrypted centroid with the other owners. In [7] a similar mechanism is proposed except that only one of the users calculates the global centroids and then shares this. Using MPC data confidentiality is preserved since data is stored locally and is not shared with other parities. However, data owners are expected to do much of the work and are thus required to have the expertise and resource to carry out the desired clustering.

Using PTC most of the processing is conducted by a third party, but with recourse to the data owner (or owners) for the similarity calculations required as the k-means clustering proceeds. A disadvantage is thus that typically a great many of such calculations will be required. Using PTC data is encoded using some form of HE and sent to a third party where the homomorphic properties of the encryption are used to manipulate the data so far as possible (centroid calculation, data aggregation, and so). For example in [5] a PTC setting is used to cluster data belonging to social network users into k clusters. The data is passed to a "Semi Honest" third party where the clustering is performed, whilst similarity is determined with reference to randomly selected users. The mechanism proposed in this paper falls into the second category, but seeks to significantly minimise the interaction with the data owner(s).

There has also been work directed at incorporating the idea of order preservation into homomorphic encryption schemes to support secure k-means clustering. One example can be found in [11] where the authors describe a mechanism whereby partial order preservation can be included in Liu's encryption scheme [10]. Using this mechanism the third party uses dynamic trap-doors that are calculated on each iteration to convert cypher-text to order preserving text1. However, data owner participation is still significant because the trapdoors need to be recalculated on each iteration. Liu's scheme as used in [11], is also the encryption scheme adopted with respect to the work presented in this paper, because of its particular homomorphic properties.

Chapter 3

Preliminaries

Before we go in depth with our scheme the reader needs to have a basic idea of some concepts which are going to be used hereafter. Hence we give a brief description of each of them.

3.1 K-Means clustering

K-means is an iterative clustering algorithm where n data items are grouped into k clusters [12]; k is specified by the user. Each cluster is represented by its centroid. On the first iteration the first k records are usually used as the centroids. The remaining records are then assigned to clusters according to the shortest "distance" from each record to each centroid. At the end of the first iteration the centroids are recalculated; this is typically, but not necessarily, done using the mean values of the attribute values for the records present in each cluster. The algorithm then again assigns the records to the clusters and continues in this iterative manner until the centroids become fixed.

3.2 Liu's Homomorphic Encryption (LHE)

In this section, we briefly describe the encryption scheme [?] [?] we use. As mentioned before we need to encrypt the data so as the curious CSP is not able to look into it as well as he must be able to perform the computations required for returning search results to the users. Homomorphic encryption is one of the most suitable encryption schemes one can use for cloud-based scenario as there is no need of generating a key for the CSP. The CSP not even getting a hint of what is inside the database can operate on it as if it were plain-text. The problem with full-scale homomorphic encryption scheme is that it involves repetitive expensive pre-processing computations which makes it inefficient for practical applications. The solution we propose is to use a light-weight homomorphic encryption scheme. The encryption schemes of this kind do provide support for simple algebraic computations of the data as well as are efficient for practical applications with good usability features. The encryption we use here is secure from cipher-text-only attacks and is suitable for a database consisting of numerical entries.

3.2.1 Key generation

This encryption scheme involves only a single key (K(m)) for encryption which is a tuple of three components, $\mathrm{K(m)} = (\Gamma, \Theta, \Phi)$. The first component Γ is a list of $[(k_1, s_1, t_1).,(k_m, s_m, t_m)]$ where k_i, s_i and t_i are random real numbers. The real numbers used in this scheme are uniformly sampled random real numbers with finite precision. To ensure that the scheme works every time correctly we have to make sure$m \geq 4$, $k_i \neq 0$ for $1 \leq i \leq m-1$, $k_m + s_m + t_m \neq 0$ and $t_i \neq 0$ for $1 \leq i \leq m-1$.

The component Θ is a pair of two random real numbers θ_1 and θ_2. The component Φ is also defined as a list, $\Phi = (\Phi_1, \Phi_2, \Phi_2)$ where Φ_j is a m dimensional vector $(\phi_{\mathrm{j1}}, ., \phi_{\mathrm{jm}})$ for $1 \leq j \leq 3$. The following section describes the way to compute the value of each $(\phi_{\mathrm{j1}}$.

- Uniformly sample m random real numbers $r_{\mathrm{j1}}, ..., r_{\mathrm{jm}}$. The numbers can be arbitrarly large.

- Compute $\phi_{\text{j1}} = k_1 * t_1 * \theta_j + s_1 * r_{\text{jm}} + k_1 * (r_{\text{j1}} - r_{\text{j(m-1)}})$
- Compute $\phi_{\text{ji}} = k_i * t_i * \theta_j + s_i * r_{\text{jm}} + k_i * (r_{\text{ji}} - r_{\text{j(i-1)}})$ for $2 \leq i \leq m-1$
- Compute $\phi_{\text{jm}} = (k_m + s_m + t_m) * r_{\text{jm}}$

3.2.2 Encryption

Once the key K(m), with $\Phi = (\Phi_1, \Phi_2, \Phi_3)$ is generated, then the encryption algorithm denoted by Enc(K(m),v) encrypts a numerical value v and gives the following output $(e_1, e_2, , e_m)$, by using the following method.

- Uniformly sample 2 arbitrarily large random real numbers ru_1 and ru_2
- Compute $ru_3 = v - ru_1 * \theta_1 ru_2 * \theta_2$
- Compute $e_i = ru_1 * \phi_{\text{1i}} + ru_2 * \phi_{\text{2i}} + ru_3 * \phi_{\text{3i}}$

3.2.3 Decryption

Now for the decryption part the algorithm denoted by Dec(K(m), $(e_1, , e_m)$) gives the plaint text v for the given cipher-text $(e_1, .., e_m)$ in the following steps.

- T = $\sum_{i=1}^{m-1} t_i$
- S = $e_m/(k_m + s_m + t_m)$
- v = $(\sum_{i=1}^{m-1}(e_i S * s_i))/T$

The conditions $m \geq 4$, $k_i \neq 0$ for $1 \leq i \leq m-1$, $k_m + s_m + t_m \neq 0$ and $t_i \neq 0$ for $1 \leq i \leq m-1$ ensure the validity of the above decryption scheme. As mentioned earlier the above scheme is secure from cipher-text only attacks and is also semantically secure as long as the plain length of the plain-texts is bounded.

Chapter 4

Proposed Method

This section presents the proposed UDM based on secure k-means clustering process. The process has two parts, a data preparation part and a clustering part. The first is conducted by the data owner and is detailed in Sub-section 5.1, while the second is conducted by the third party and is detailed in Sub-section 5.2. Note that the proposed process delegates some of the processing to the data owners, but this is limited to the generation of SMs and is significantly less than in the case of more conventional PTC approaches such as that presented in [5].

4.1 Data Owner Process

Data sets are prepared for outsourcing by generating a UDM and translating the raw data into a suitable encrypted format. The pseudo code for the process is given in Algorithm 3. The input to the algorithm is a dataset D, a set of attributes A that are featured in D and the number of required sub-cyphers m, the later required for Liu's encryption scheme adopted with respect to the work presented in this paper. Note that k-means clustering works only on numerical (or pseudo numerical) data, because similarity measurements are central to the operation of the mechanism. Therefore categorical (or labelled) data needs to be replaced with discrete integers values before any further processing can be conducted, this is done in line 2 of the algorithm. A list of secret keys K(m) is then defined (line 3). The processed data set is then encrypted to form a cypher data set D0 using Algorithm 1 (lines 5 to 7). Next the desired UDM, U, is dimensioned (line 8) and populated (lines 9 to 15); recall from Section 4 that vxz is the value of the zth attribute in the feature vector describing the xth record in the dataset, while vyz is the value of the zth attribute in the feature vector describing the yth record in the dataset. On completion (line 16) the process returns the encrypted data set D0 and the generated UDM U ready to be sent to the third party data miner.

4.2 Third Party Process

The secure k-means clustering is conducted by the third party data miner follow- ing a processes very similar to the traditional k-means algorithm as described in Sub-section 3.1. The pseudo code presented in Algorithm 4 describes this process. The input is the encrypted data set D0, the UDM U (previously sub- mitted to the third party) and the number of desired clusters k. We commence by dimensioning the set of clusters C = fC1;C2; : : : ;Ckg and assigning the first k encrypted records from D0 to it (lines 2 and 3). We then define a second set Cent = fc1; c2; : : : ; ckg to hold the current centroids (line 4). Next the re- mainder of D0 is processed and assigned to clusters using the populateClusters sub-process (line 5) given at the end of the algorithm. Records are assigned to clusters according to the similarity between the record and centroids as calcu- lated using the UDM U as follows (where rx is record x 2 D, and ry is record y 2 D representing cluster centroid y (1 y k): sim(U; rx; ry) = zX=jAj z=1 U[x;y;z] (2) We then calculate the new centroids (line 6). Next we enter into a loop (lines 7 to 14) which repeats until Cent and Cent0 are the same. At the start of each iteration we create an encrypted SM S0 (line 8). However, to update U we need real values, the content therefore needs to be decrypted. This can only be done by the owner (line 9). Note that (not shown in the algorithm) when updating U we only need to update the records in the second dimension representing cluster centroids. With the new centroids we again assign all records to C using the populateClusters sub-process (line 12) in the same manner as before. We continue in this manner till the process terminates.

Chapter 5

Security Analysis

In this section we present the security analysis of our proposed scheme based on the parameters of confidentiality. The data must be safe from insider as well as outsider attacks. We try to achieve immunity from both type of attacks.

5.1 Confidentiality

One of the key parameters in any outsourced database architecture, it must always be ensured. Majority of the databases around the globe are filled with personal data of organizations and individuals and if leaked can easily aid identity theft on a large scale. Hence it is the one the most important metric to evaluate any data outsourcing paradigm.

Theorem *The proposed scheme is secure in data confidentiality.*

The third party neither needs any part of the plaintext values nor the encryption key to do its part of computations and return the search results. The third party never actually has the plain text and can never know what is inside the database. Hence the proposed scheme is secure from a curious server.

Moreover, the scheme we propose uses homomorphic encryption and is secure from cipher text-only attacks. So that neither the CSP nor any hacker, intercepting the packets between CSP and users which contain only the ciphertexts, can crack the encryption. Hence we can say that proposed scheme is secure in data confidentiality.
From the above discussion it is evident that our proposed scheme reaches the desired level of security and performs well on all metrics including confidentiality and integrity of the database. It is lightweight yet proven to be secure from cipher-text only attack.Our scheme can achieve the corresponding level of confidentiality if the encryption key is strictly limited to the dataowner and authorized users only. It is assumed that the authorization between data owners and users is appropriately done. That is, any unauthorized users or revoked ones must not get the valid encryption key.A feasible optimization such as adding message authenticators adds additional functionalities such as Proof of Retrievability (PoR). Also the databse becomes immune from deletion and modification attacks.

Chapter 6

Performance Analysis

This section of the report includes the experimental evaluation of the proposed scheme. The local machine we used had python 3.6 running on Intel Core-i5 5200U 2.20 GHz CPU with 8 GB of memory.

The evaluation of the proposed method is presented in this section using ten NSL-KDD dataset for intrusion detection. Table 1 and Table 2 gives some statistical information concerning this data set.

Training Dataset (20 %)	Attack Type (21)
DoS	Back,Land,Neptune,Pod,Smurf,teardrop
Probe	Satan,Ipsweep,Nmap, Portsweep
R2L	Guess_Password,Ftp_write,Imap, Phf,Multihop,Warezmaster,Warezclient, Spy
U2R	Buffer_overflow, Loadmodule, Rootkit

Table 6.1: Attacks in Training Dataset

Total Attributes			
Attribute	type	Attribute	type
duration	continuous	is_guest_login	continuous
protocol_type	symbolic	srv_count	continuous
service	symbolic	serror_rate	continuous
flag	symbolic	srv_serror_rate	continuous
src_bytes	continuous	rerror_rate	continuous
dst_bytes	continuous	srv_rerror_rate	continuous
land	continuous	same_srv_rate	continuous
wrong_fragment	continuous	diff_srv_rate	continuous
urgent	continuous	srv_diff_host_rate	continuous
hot	continuous	dst_host_count	continuous
num_failed_logins	continuous	dst_host_srv_count	continuous
logged_in	continuous	dst_host_same_srv_rate	continuous
num_compromised	continuous	dst_host_diff_srv_rate	continuous
root_shell	continuous	dst_host_same_src_port_rate	continuous
su_attempted	continuous	dst_host_srv_diff_host_rate	continuous
num_root	continuous	dst_host_serror_rate	continuous
num_file_creations	continuous	dst_host_srv_serror_rate	continuous
num_shells	continuous	dst_host_rerror_rate	continuous
num_access_files	continuous	dst_host_srv_rerror_rate	continuous
num_outbound_cmds	continuous	is_host_login	continuous

Table 6.2: List of attributes with their type

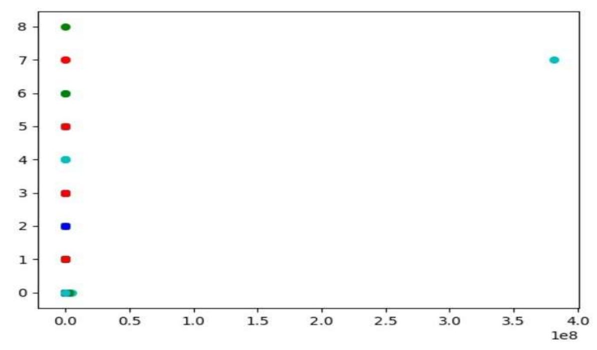
The Data sets have integer and categorical attribute types. The number of classes of attacks used as the value for k and the class column omitted from the data set. The proposed process was implemented using the Python programming language. Cluster configuration correctness was measures by comparing the results obtained with results obtained using standard (unencrypted) k-means clustering. The metric used was the sum of squared errors(SSE) within clusters, calculated using

$$SSE = \sum_{j=1}^{k} \sum_{i=1}^{n} (x_i - \bar{x})^2$$

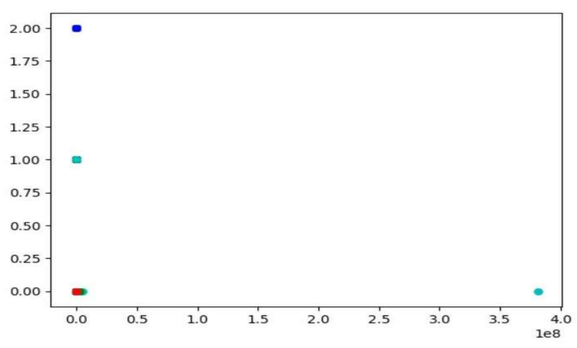
The results are presented in the following graphs of attribute vs attribute and Table 3, the runtime is the time to complete the clustering excluding encryption and UDM generation. From the table it can be seen firstly that the cluster configurations produced using the proposed secure k-means clustering were similar to those produced using standard k-means clustering as evidenced by mean squared error. Thus it can be concluded that the homomorphic properties used to calculate the centroids and distances between centroids do not effect the accuracy of the clustering. In terms of run time the slightly more number of iterations were required. The overall run time required for the secure k-means approach to produce the desired cluster configurations, as expected, was longer than in the case of standard k-means clustering. Note that the time complexity for generating an initial UDM will be in the order of O(DXCXk). The time to decrypt a SM, on each iteration, will be negligible. However, it should be noted that with respect to the experiments reported here the data owner and third party were both hosted on the same machine, thus there was no "message passing" overhead; in "real life" the data owner and third party will be hosted on separate machines, time for message passing would add to the run time although not significantly.

Training Dataset	No of iterations	Mean squared error within clusters
NSL-KDD Dataset	11	32191.588778718917

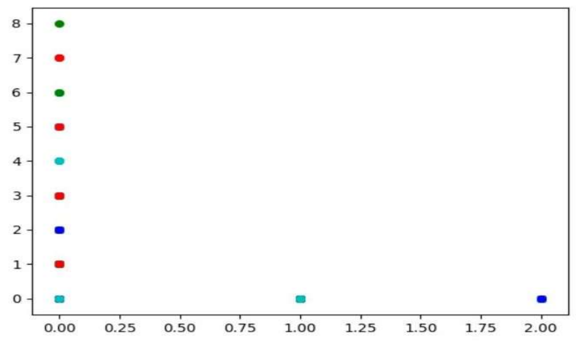
Table 6.3: Results Summary



(a) flag vs source bytes



(b) source bytes vs protocol type



(c) protocol type vs flag

Figure 6.1: Clustering results visualized on axes of attributes

Chapter 7

Conclusion and Future Work

In this report we mainly present an efficient implementation of secure K-means clustering using homomorphic encryption. The scheme we implement has proven its worth by performing well on metrics of clustering and being efficient as well. The scheme being homomorphic in nature is more easy to maintain as it doesn't involve any key for the third party. Moreover, the third party can perform all sorts of computations such as subtraction, addition, mean calculation as it is basically operating as if on a plain-text database.

The scheme that we present is extremely efficient and yet there is scope for improvement such as reducing the time for iterations or limiting the user-server interaction. Moreover, the theme of our B.Tech project is *"Blind Quantum Machine Learning"*. We have focused on the blind part till now and in the next semester we will be focusing the quantum aspect of the machile learning algorithm.

Bibliography

[1] Rakesh Agrawal and Ramakrishnan Srikant. Privacy-preserving data mining. SIGMOD Rec., 29(2):439-450, May 2000.

[2] Scott Berinato. Theres no such thing as anonymous data. Harvard Business Review, February, 2015.

[3] T. Chen, J. Chen, and B. Zhou. A system for parallel data mining service on cloud. In 2012 Second International Conference on Cloud and Green Computing, pages 329-330, November 2012.

[4] Hitesh Chhinkaniwala and Sanjay Garg. Privacy preserving data mining techniques: Challenges and issues. In Proceedings of International Conference on Computer Science Information Technology, CSIT, page 609, July 2011.

[5] Zekeriya Erkin, Thijs Veugen, Tomas Toft, and Reginald L Lagendijk. Privacy-preserving user clustering in a social network. In 2009 First IEEE International Workshop on Information Forensics and Security (WIFS), pages 96-100. IEEE, December 2009.

[6] Oded Goldreich. Foundations of cryptography: volume 2, basic applications. Cambridge university press, 2009.

[7] Somesh Jha, Luis Kruger, and Patrick McDaniel. Privacy preserving clustering. In European Symposium on Research in Computer Security, pages 397-417. Springer, September 2005.

[8] M. Lichman. UCI machine learning repository, 2013.

[9] Yehuda Lindell and Benny Pinkas. Privacy preserving data mining. Journal of cryptology, 15(3):177-206, June 2002.

[10] Dongxi Liu. Homomorphic encrypton for database querying, December 2013.

[11] Dongxi Liu, Elisa Bertino, and Xun Yi. Privacy of outsourced k-means clustering. In Proceedings of the 9th ACM symposium on Information, computer and communications security, pages 123-134. ACM, June 2014.

[12] James MacQueen. Some methods for classification and analysis of multivariate observations. In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, pages 281-297. Oakland, CA, USA., June 1967.

[13] Deepti Mittal, Damandeep Kaur, and Ashish Aggarwal. Secure data mining in cloud using homomorphic encryption. In Cloud Computing in Emerging Markets (CCEM), 2014 IEEE International Conference on, pages 1-7. IEEE, October 2014.

[14] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics, 20:53-65, November 1987.

[15] Meena Dilip Singh, P Radha Krishna, and Ashutosh Saxena. A privacy preserving jaccard similarity function for mining encrypted data. In TENCON 2009-2009 IEEE Region 10 Conference, pages 1-4. IEEE, January 2009.

[16] Jaideep Vaidya, Christopher W Clifton, and Yu Michael Zhu. Privacy preserving data mining, volume 19. Springer Science Business Media, 2006.

[17] Yang Yang and MA Maode. Semantic searchable encryption scheme based on lattice in quantum-era. Journal of Information Science Engineering, 32(2):425 438, March 2016.

[18] Coenen, FP , Almutairi, and Dures, K (2017) K-Means Clustering Using Homomorphic Encryption and an Updatable Distance Matrix: Secure Third Party Data Clustering with Limited Data Owner Interaction. In: 19th International Conference on Big Data Analytics and Knowledge Discovery, 2017-08-28 - 2017-05-31, Lyon, France.