Metadata of the chapter that will be visualized in SpringerLink

Book Title	Cyberspace Safety and Security	
Series Title		
Chapter Title	A Secure Density Peaks Clustering Algorithm on Cloud Computing	
Copyright Year	2019	
Copyright HolderName	Springer Nature Switzerland AG	
Author	Family Name	**Ci**
	Particle	
	Given Name	**Shang**
	Prefix	
	Suffix	
	Role	
	Division	School of Computer and Information
	Organization	Anhui Normal University
	Address	Wuhu, China
	Division	
	Organization	Anhui Provincial Key Laboratory of Network and Information Security
	Address	Wuhu, 241002, China
	Email	cs_xxy1994@163.com
Corresponding Author	Family Name	**Sun**
	Particle	
	Given Name	**Liping**
	Prefix	
	Suffix	
	Role	
	Division	School of Computer and Information
	Organization	Anhui Normal University
	Address	Wuhu, China
	Division	
	Organization	Anhui Provincial Key Laboratory of Network and Information Security
	Address	Wuhu, 241002, China
	Email	slp620@163.com
Author	Family Name	**Liu**
	Particle	
	Given Name	**Xiaoqing**
	Prefix	
	Suffix	
	Role	
	Division	School of Computer and Information
	Organization	Anhui Normal University
	Address	Wuhu, China

	Division	
	Organization	Anhui Provincial Key Laboratory of Network and Information Security
	Address	Wuhu, 241002, China
	Email	xqliu7788@163.com
Author	Family Name	**Du**
	Particle	
	Given Name	**Tingli**
	Prefix	
	Suffix	
	Role	
	Division	School of Computer and Information
	Organization	Anhui Normal University
	Address	Wuhu, China
	Division	
	Organization	Anhui Provincial Key Laboratory of Network and Information Security
	Address	Wuhu, 241002, China
	Email	2260436487@qq.com
Author	Family Name	**Zheng**
	Particle	
	Given Name	**Xiaoyao**
	Prefix	
	Suffix	
	Role	
	Division	School of Computer and Information
	Organization	Anhui Normal University
	Address	Wuhu, China
	Division	
	Organization	Anhui Provincial Key Laboratory of Network and Information Security
	Address	Wuhu, 241002, China
	Email	zxiaoyao@ahnu.edu.cn

Abstract	Cloud computing provides users with the convenience of data outsourcing computing at risk of privacy leakage, and clustering algorithms have high computational overhead when dealing with large datasets. Aiming at the above problems, this paper presents a security density peak clustering algorithm based on grid in hybrid cloud environment. First, the client uses the homomorphic encryption method to build the encrypted objects with user datasets. Second, the client uploads the encrypted objects to the cloud servers to implement the security protocols proposed in this paper. Finally, the cloud servers return the perturbation clustering results to the client to eliminate the disturbance. In the proposed scheme, only encryption and removing perturbation are performed on the client, ensuring that the client has lower computational complexity. Security analysis and experimental results show that the scheme proposed in this paper can improve the efficiency and accuracy of clustering algorithm under the premise of protecting user privacy.

Keywords	Cloud computing security - Density peaks clustering algorithm - Data mining - Privacy preserving - Homomorphic encryption

A Secure Density Peaks Clustering Algorithm on Cloud Computing

Shang Ci[1,2], Liping Sun[1,2(✉)], Xiaoqing Liu[1,2], Tingli Du[1,2], and Xiaoyao Zheng[1,2]

[1] School of Computer and Information, Anhui Normal University, Wuhu, China
cs_xxyl994@163.com, slp620@163.com, xqliu7788@163.com, 2260436487@qq.com, zxiaoyao@ahnu.edu.cn

[2] Anhui Provincial Key Laboratory of Network and Information Security, Wuhu 241002, China

Abstract. Cloud computing provides users with the convenience of data outsourcing computing at risk of privacy leakage, and clustering algorithms have high computational overhead when dealing with large datasets. Aiming at the above problems, this paper presents a security density peak clustering algorithm based on grid in hybrid cloud environment. First, the client uses the homomorphic encryption method to build the encrypted objects with user datasets. Second, the client uploads the encrypted objects to the cloud servers to implement the security protocols proposed in this paper. Finally, the cloud servers return the perturbation clustering results to the client to eliminate the disturbance. In the proposed scheme, only encryption and removing perturbation are performed on the client, ensuring that the client has lower computational complexity. Security analysis and experimental results show that the scheme proposed in this paper can improve the efficiency and accuracy of clustering algorithm under the premise of protecting user privacy.

Keywords: Cloud computing security · Density peaks clustering algorithm · Data mining · Privacy preserving · Homomorphic encryption

1 Introduction

With the rapid development of information technology, such as cloud computing, Internet of Things and social networks, industrial Internet of Things (IIoT) has led the era of intelligent enterprises and industries [1]. As an important research field of data mining, clustering aims at assigning objects to different junior high schools according to similarity. Big data usually contains a large number of samples and has very high dimensional attributes, which has high computational complexity in cluster analysis. Today, more and more enterprises are storing data in cloud servers, and their powerful computing power makes it easy to process big data [2, 3].

Because cloud service providers can be malicious, users' privacy may be compromised when sensitive data is outsourced directly to cloud servers for computation [4, 5]. In order to protect the security of user data in the cloud servers, the client encrypts the private data before outsourcing. That is, the client encrypts the data and

J. Vaidya et al. (Eds.): CSS 2019, LNCS 11982, pp. 1–9, 2019.
https://doi.org/10.1007/978-3-030-37337-5_43

outsourcing it to the cloud server. The cloud server calculates it directly on the ciphertext, and the cloud server returns it to the client, who decrypts it. During this process, the cloud server does not learn the middle of user sensitive data and computation, thus the security can be guaranteed. Zhang *et al*. [6] presented a privacy preserving HOCFS (PPHOCFS) method utilizing BGV encryption scheme. However, because some operations cannot be realized, such as comparison and division, only the similarity is computed on ciphertext, whereas CFS is still calculated on plaintext. To protect the original data stored in the cloud, Liu *et al*. [7] proposed a privacy-preserving *K*-means clustering algorithm using its own homomorphic cryptosystem for outsourced databases. It can preserve both data privacy and query privacy, but does not protect data access patterns. Rao *et al*. [8] proposed the privacy-preserving *K*-means clustering algorithm using the Paillier cryptosystem that can guarantee the confidentiality of the outsourced databases. However, it requires a high computation cost due to the usage of a bit array-based comparison.

Aiming at aforementioned challenges, a security density peak clustering algorithm (SDPC) based on grid in hybrid cloud environment is presented in this paper. The clustering centers are quickly found through the idea of grid, and the efficiency of density peak clustering algorithm is improved. At the same time, to ensure the security of user data, the client uses the homomorphic encryption scheme to encrypt the privacy data and upload it to the cloud server. Using public and private cloud operations reduces user computing overhead. Public and private clouds make up a hybrid cloud with public cloud computing capabilities and private cloud security. This paper uses the Paillier cryptosystem, the private cloud generates the public key *pk* and the private key *sk*, and publishes the *pk* to the users and the public cloud. Users use *pk* to encrypt private data, while the public cloud carries out secure clustering operations. In practical applications, public cloud providers are usually well-established IT companies like Microsoft and Google, whereas private cloud providers are usually special institutions under the government supervision. For the sake of reputation and commercial interests, a collusion between them is highly unlikely and they will not maliciously steal user information.

2 Preliminaries

The symbols used in this paper and their semantic meanings are as follows. n: number of samples; q: dimension of samples; a: original data; $[[a]]$: encrypted data; μ: edge length of grid; $[[\mu]]$: encrypted edge length of grid; ρ_i: local density of sample i; δ_i: distance from sample i to the local density is larger than its nearest sample j; $[[\rho_i]]$: encrypted local density of sample i; $[[\delta_i]]$: encrypted distance from sample i to the local density is larger than its nearest sample j; α: probability parameter; λ: magnification factor.

This paper used the Euclidean distance as the criterion for representing distance of sample points $a_i = (a_{i1}, a_{i2}, \ldots, a_{iq})$ and $a_j = (a_{j1}, a_{j2}, \ldots, a_{jq})$, where $1 \leq i \leq n$, $1 \leq j \leq n$.

$$d_{(i,j)} = [(a_{i1} - a_{j1})^2 + (a_{i2} - a_{j2})^2 + \ldots + (a_{iq} - a_{jq})^2]^{\frac{1}{2}} = [\sum_{m=1}^{q} (a_{im} - a_{jm})^2]^{\frac{1}{2}} \quad (1)$$

Definition 1. Assume there is a dataset with size $n \times q$ and the data of the ith, mth dimension is a_{im}, where $1 \leq i \leq n$, $1 \leq m \leq q$. We partition the data space by dividing each dimension into equal and disjoint grid cells and the edge length μ of each grid is defined as follows:

$$\mu = \alpha \left(\prod_{m=1}^{q} \frac{a_{1m} + a_{2m} + \ldots + a_{nm}}{n} \right)^{\frac{1}{q}} \quad (2)$$

α is the parameter used to adjust the edge of each grid and n is the number of the data points in the dataset.

Definition 2. Assume there is a q-dimensional dataset $X = \{x_1, x_2, \ldots, x_n\}$ and the data space is divided into $\{\theta_1, \theta_2, \ldots, \theta_k\}$ grid cells according to **Definition 1**. Then the data points are mapped into corresponding grid cells, the density of the cell is:

$$\rho_{\theta_i} = count(G_{\theta_i}) \quad (3)$$

Where function count() represents the number of points in the gird cell whose grid number is G_{θ_i}.

3 Basic Security Primitives

3.1 Existing Security Protocol

The existing security protocols involved in SDPC are listed in Table 1, including secure multiplication (SM) [9], secure comparison (SC) [10], secure division 1 (SD1) [11] and secure sort (SSOAT$_k$) [12].

Table 1. Table of existing security protocol.

Protocol	Definition
Secure multiplication	$SM([[a]], [[b]]) \rightarrow [[a \cdot b]]$
Secure comparison	$SC([[a]], [[b]]) \rightarrow [[a \geq b]]$
Secure division 1	$SD1([[a]], b) \rightarrow [[qu_1]]$
Secure sort	$SSOAT_k([[a_1]], \ldots, [[a_n]], k) \rightarrow ([[a'_1]], \ldots, [[a'_k]])$

3.2 The Proposed Protocol

In order to implement the algorithm, a set of protocols is proposed as a standard. In addition, in order to control floating point precision, an amplification factor is added to the protocol. C_1 and C_2 denote the public cloud and private cloud, respectively.

(1) **SP Protocol:** As there is no homomorphic exponentiation operations in the Paillier cryptosystem, it can not directly support the secure computation between objects. In view of the above problems, this paper proposes a safe exponentiation method, the whole process is shown as Algorithm 1.

Algorithm 1: SP Protocol.

Input: C_1 has $[[\lambda_p a]]$ and b, C_2 has sk
Output: Encrypted exponential result $[[\lambda_p a^b]]$ only to C_1

C_1
1: Select the top k maxima $m_1, m_2, ..., m_k$ from all possible values of b
2: $i \leftarrow 1$
3: **Repeat**
4: Compute $[[\lambda_p a^{m_i}]]$, $[[\lambda_p m_i]]$
5: $i \leftarrow i+1$
6: **Until** $i > k$
C_1, C_2
7: $i \leftarrow 1$
8: **Repeat**
9: $[[c_i]] \leftarrow SC([[\lambda_p b]] , [[\lambda_p m_i]])$
10: $[[\lambda_p s_i]] \leftarrow SM([[c_i]] , [[\lambda_p a^{m_i}]])$
11: $i \leftarrow i+1$
12: **Until** $i > k$
C_1
13: $[[\lambda_p a^b]] \leftarrow \prod_{i=1}^{k} [[\lambda_p s_i]]$

(2) **SED Protocol:** In order to safely calculate the distance between objects, SED protocol is proposed, the goal of which is to calculate $[[\lambda_0 d_{(i,j)}]]$ safely. The basic idea of SED is shown in Algorithm 2.

Algorithm 2: SED Protocol.

Input: C_1 has encrypted data $[[\lambda_0 a_{im}]]$ and $[[\lambda_0 a_{jm}]]$, where $1 \le m \le q$. C_2 has sk

Output: Encrypted distance result $[[\lambda_0 d_{(i,j)}]]$ only to C_1

C_1, C_2

1: $i \leftarrow 1$
2: **Repeat**
3: $j \leftarrow 1$
4: **Repeat**
5: $[[\lambda_0(a_{im} - a_{jm})]] \leftarrow [[\lambda_0 a_{im}]] * [[\lambda_0 a_{jm}]]^{N-1}$
6: $[[\lambda_0{}^2 d_m]] \leftarrow SM([[\lambda_0(a_{im} - a_{jm})]], [[\lambda_0(a_{im} - a_{jm})]])$
7: $[[\lambda_0 d_m]] \leftarrow SD_1([[\lambda_0{}^2 d_m]], \lambda_0)$
8: $j \leftarrow j + 1$
9: **Until** $j > n$
10: $i \leftarrow i + 1$
11: **Until** $i > n$

C_1

12: $[[\lambda_0 d_{(i,j)}]] \leftarrow \sum_{m=1}^{q} [[\lambda_0 d_m]]$

4 Algorithm Description

4.1 Security Density Peak Clustering Algorithm Based on Grid

This section describes secure density peak clustering algorithm (SDPC) grid-based in hybrid clouds. By using the algorithm proposed in this paper, cloud computing is used to securely provide high quality clustering services without revealing any private information.

In this algorithm, C_1 holds private input $[[a_1]], \ldots, [[a_n]]$, and C_2 holds sk. The goal of the SDPC is to compute encrypted clustering result $[[cl]]$ without revealing any information about n objects to C_1 and C_2. At the end, only C_1 knows the final result $[[cl]]$.

Algorithm 3: SDPC.

Input: C_1 has encrypted data $[[a_1]],...,[[a_n]]$, C_2 has sk, the screening ratio: r.

Output: Encrypted clustering result $[[cl]]$ only to C_1.

C_1, C_2

1: Calculate the distance $[[d_{(i,j)}]]$ from all the encrypted data
2: Calculate the edge length μ of each grid
3: Map the encrypted data $[[a_1]],...,[[a_n]]$ into the corresponding grid cells which taken by **Definition 1**
4: Count the density ρ_{θ_i} for each grid cell accoring to formula (3) and sort them according to secure sort protocol
5: Screen the encrypted data in first r% 'dense' grids based on the screening ratio r, and remove the other encrypted data points to form a new encrypted dataset $A = \{A_1, A_2, ..., A_t\}$ for finding cluster centers
6: Calculate the $[[\rho_i]]$ and $[[\delta_i]]$ for each data point in dataset A according to $[[d_{(i,j)}]]$
7: Select cluster centers with larger $[[\rho_i]]$ and $[[\delta_i]]$ based on decision graph
8: Assign the remaining data points in dataset A to the class of nearest point with equal or higher density
9: Assign the $n-t$ data points which removed in step 5 to the nearest classes according to the 'nearest neighbor' principle
10: Return the encrypted clustering result $[[cl]]$

4.2 Security Analysis

The security of the SDPC method can be proved by using the semi-honest model in secure two-party computation, and the users are not involved in the specific calculation of the algorithm. Both the public cloud C_1 and the private cloud C_2 in the algorithm follow the rules of each protocol, but they all try to infer the user's private information

during the execution of the protocols. Since all intermediate and final results of the algorithm are protected using the formal Paillier cryptosystem, C_1 cannot obtain any private information. At the same time, the output of each protocol is a ciphertext that only C_1 knows. In addition, although C_2 can use the private key sk to decrypt intermediate results, it can only see random values or disturbed user data. Because each step of the protocols in this paper uses homomorphic encryption attribute or properly verified security classic protocols, it is claimed that the proposed SDPC method is completely secure based on the composition theorem [13].

4.3 Complexity Analysis

This section combines the characteristics of cloud computing cost and semi-integrity hybrid cloud security framework to theoretically analyze the computational and communication costs of the proposed scheme. Let the number of elements of a dataset be m, where the number of zero elements is m_0, while the number of nonzero elements is m_1, and the number of the object be n.

Computation Cost: The computational complexity of the client is $O(m_1 n)$.

According to the SDPC, the computation cost of the cloud T consists of the cost of secure computing SED and the cost of SDPC T_{SDPC}, which is defined as formula (4).

$$T = T_{SED} + T_{SDPC} \tag{4}$$

where time complexity T_{SED} is $O((m^2 - m_0^2)n^2)$, and time T_{SDPC} complexity is $O(rn^2 + k(n-t))$. Therefore, the total computation T is $O((m^2 - m_0^2)n^2 + rn^2 + k(n-t))$.

5 Performance Evaluation

For performance analysis, we do our experiment on Intel Xeon CPU E5-2620, 2.0 GHz and 4-GB physical memory. We considered the performance and time efficiency of the typical clustering algorithm and the comparability of this study, and compared and analysed the proposed SDPC algorithm with K-means algorithm [14] and DPC algorithm [15]. The datasets used for our experiment from the UCI machine learning library. The first Iris dataset consists of 150 records with 4 attributes. The second Adult dataset consists of 32561 records with 14 attributes.

In order to measure the influence of dataset size on running time, the client encrypts a quarter, half, three-fourth, and all of the samples of the datasets. The running time is shown in the Fig. 1. The running time of the SDPC algorithm is lower than K-means algorithms, and no significant increase is observed when compared with the original DPC algorithm.

For the two aforementioned datasets, the experimental results of SDPC algorithm and the comparison algorithm are shown in Table 2. As shown in Table 2, by using two evaluation metrics ACC and F-measure, to evaluate the clustering results, our proposed algorithm outperforms other algorithms on an average; this shows that SDPC algorithm produces more accurate clustering centers. For the Iris dataset, the ACC

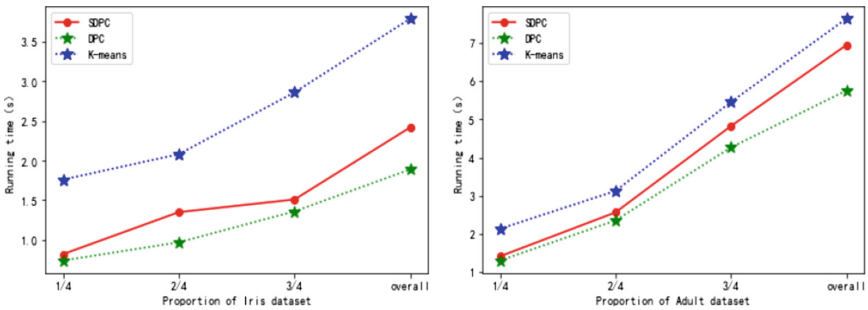

Fig. 1. Running times of comparison algorithms on two datasets.

index of SDPC algorithm is 52.6% higher than that of DPC algorithm. For the Adult dataset, the F-measure index of SDPC algorithm is 41.7% higher than that of *K*-means algorithm.

Table 2. Experimental results of different algorithms on datasets.

Datasets	Algorithms	ACC	F-measure
Iris	*K*-means	0.793	0.783
	DPC	0.576	0.712
	SDPC	0.879	0.867
Adult	*K*-means	0.653	0.521
	DPC	0.694	0.614
	SDPC	0.801	0.738

6 Conclusion

Aiming to provide clustering service for big data mining applications securely and efficiently, this paper proposes a secure density peak clustering algorithm grid-based in hybrid clouds. In this algorithm, all computing tasks are performed on the cloud without exposing or inferring any sensitive information, and clustering centers can be quickly found. This method not only improves efficiency, but also preserves user privacy. In the end, the performances of the proposed SDPC method are evaluated on two datasets in terms of clustering accuracy and efficiency. In our future studies, we will consider the secure method of other clustering algorithms and apply them to practical problems.

Acknowledgment. This work is supported by the National Natural Science Foundation of China under Grant 61602009 and Grant 61672039, and the Anhui Provincial Natural Science Foundation of China under Grant 1808085MF172.

References

1. Yin, S., Kaynak, O.: Big data for modern industry: challenge and trends. Proc. IEEE **103**(2), 143–146 (2015)
2. Armbrust, M., Fox, A., Griffith, R.: A view of cloud computing. Commun. ACM **53**(4), 50–58 (2010)
3. Fang, S.: An integrated approach to snowmelt flood forecasting in water resource management. IEEE Trans. Ind. Inform. **10**(1), 548–558 (2014)
4. Ma, M., He, D., Kumar, N., Choo, K.K., Chen, J.: Certificateless searchable public key encryption scheme for industrial internet of things. IEEE Trans. Ind. Inform. **14**(2), 759–767 (2018)
5. Esposito, C., Castiglione, A., Martini, B., Choo, K.K.: Cloud manufacturing: security, privacy, and forensic concerns. IEEE Cloud Comput. **3**(4), 16–22 (2016)
6. Zhang, Q., Yang, L.T., Chen, Z., Fan, Y.B.: PPHOCFS: privacy preserving high-order CFS algorithm on the cloud for clustering multimedia data. ACM Trans. Multimed. Comput. Commun. Appl. **12**(4), 66:1–66:15 (2016)
7. Liu, D., Bertino, E., Yi, X.: Privacy of outsourced k-means clustering. In: Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security (ICCS), pp. 123–134 (2014)
8. Rao, F.-Y., Samanthula, B.K., Bertino, E., Yi, X., Liu, D.: Privacy-preserving and outsourced multi-user k-means clustering. In: IEEE Conference on Collaboration and Internet Computing (CIC), pp. 80–89 (2015)
9. Samanthula, B.K., Elmehdwi, Y., Jiang, W.: K-nearest neighbor classification over seman tically secure encrypted relational data. IEEE Trans. Knowl. Data Eng. **27**(5), 1261–1273 (2015)
10. Bost, R., Popa, R.A., Tu, S., Goldwasser, S.: Machine learning classification over encrypt-ed data. In: Proceedings of 22nd Annual Network and Distributed System Security Symposium, pp. 8–11 (2015)
11. Veugen, T.: Encrypted integer division and secure comparison. Int. J. Appl. Crypt. **3**(2), 166–180 (2014)
12. Zhao, Y.L., Yang, L.T., Sun, J.Y.: A secure high-order CFS algorithm on clouds for industrial internet of things. IEEE Trans. Ind. Informat. **14**(8), 3766–3774 (2018)
13. Goldreich, O.: Foundations of Cryptography. Basic Applications. Cambridge University Press, Cambridge (2004)
14. Macqueen, J.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability (BSMSP), pp. 281–297 (1967)
15. Rodriguez, A., Laio, A.: Clustering by fast search and find of density peaks. Science **344** (6191), 1492–1496 (2014)