IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Issue-5, May 2019 E-ISSN: 2347-2693

Secured MapReduce Based K-Means Clustering In Big Data Framework

[1*]D. Saidulu, [2]V. Devasekhar, [3]V. Swathi

[1, 2, 3]Dept. of Computer Science and Engineering, Guru Nanak Institutions Technical Campus –Hyderabad, Telangana, India

DOI: https://doi.org/10.26438/ijcse/v7i5.14271430 | Available online at: www.ijcseonline.org

Accepted: 14/May/2019, Published: 31/May/2019

Abstract- Clustering is a significant task of research in data mining and analysis of statistics, which is initiate in many areas, including health care, social networking, image analysis, object recognition, etc. volume, quantity and speed. To effectively manage large-scale data sets and clusters, public cloud infrastructure plays an important role in both performance and economy. However, the use of public cloud services inevitably involves confidentiality issues. Indeed, not only a large scale of data mining applications but also deals with sensitive data such as personal healthcare information, location data, financial data, etc. In this paper we proposed Novel Secured MapReduce Based K-Means Clustering in Big Data Framework. This scheme achieves clustering speed and accuracy that are comparable to the K-means clustering without privacy protection. Furthermore we design securely integrated MapReduce framework and make it extremely suitable for parallelized processing in cloud computing environment.

***Keywords*:** K-means clustering, Data encryption, Privacy-preserving, MapReduce.

I. INTRODUCTION

Clustering techniques have been widely adopted in many real world data analysis applications, such as customer behavior analysis, targeted marketing, digital forensics, etc. With the explosion of data in today's big data era, a major trend to handle a clustering over large-scale datasets is outsourcing it to public cloud platforms. This is because cloud computing offers not only reliable services with performance guarantees, but also savings on in-house IT infrastructures. However, as datasets used for clustering may contain sensitive information, e.g., patient health information, commercial data, and behavioral data, etc., directly outsourcing them to public cloud servers inevitably raise privacy concerns. In this paper, we propose a practical privacy-preserving K-means clustering scheme that can be efficiently outsourced to cloud servers. Our scheme allows cloud servers to perform clustering directly over encrypted datasets, while achieving comparable.

Computational complexity and accuracy compared with clustering's over unencrypted ones. We also investigate secure integration of MapReduce into our scheme, which makes our scheme extremely suitable for cloud computing environment..

II. RELATED WORK

2.1 Privacy-Preserving Data Mining (PPDM)

Our work is closely related to the field of privacy-preserving data mining (PPDM) [7], [8]. Several techniques have been proposed for the clustering task under the PPDM model (e.g., [1]–[4]). However, we stress that our problem setting is somewhat different from the PPDM model. On one hand, under PPDM, each user owns a piece of dataset (typically a vertically or horizontally partitioned dataset) and the goal is for them to collaboratively perform the clustering task on the combined data in a privacy-preserving manner. On the other hand, our work is motivated by the cloud computing model where users can outsource their encrypted databases to a federated cloud environment. Under our problem setting, the federated cloud performs the clustering task over encrypted data and the users do not participate in any of the underlying computations. As a result, existing PPDM techniques for the clustering task are not applicable to the PPODC problem.

2.2 Fully Homomorphic Encryption (FHE)

A straightforward way to solve the PPODC problem is for the users to encrypt their data using a fully holomorphic encryption (FHE) scheme, e.g., [5], and outsource the encrypted data to a cloud. Here the secret key should be known only to the users (or shared among them). Since FHE allows one to perform arbitrary computations over encrypted data without decrypting the data, the cloud can perform the clustering task over encrypted data and return the encrypted clustering results to the users who can decrypt them. Though the FHE schemes enable arbitrary searches or operations over encrypted data, such techniques are very expensive and their usage in practical applications is decades away. For example, it was shown in [6] that even for weak security

parameters one bootstrapping operation of a homomorphic operation would take at least 30 seconds on a high performance machine.

III. PROBLEM STATEMENT

The problem of privacy-preserving K-means clustering has been investigated under the multi-party secure computation model, in which owners of distributed datasets interact for clustering without disclosing their own datasets to each other. In the multi-party setting, each party has a collection of data and wishes to collaborate with others in a privacy preserving manner to improve clustering accuracy. Differently, the dataset in clustering outsourcing is typically owned by a single entity, who aims at minimizing the local computation by delegating the clustering task to a third-party cloud server.

Furthermore, present multi-party designs all the time depend on prevailing but affluent cryptographic primitives to achieve collaborative secure computation among multiple parties, and are inefficient for large-scale datasets. Thus, these multi-party designs are not practical for privacy-preserving outsourcing of clustering. Another line of research that targets at efficient privacy-preserving clustering is to use distance preserving data perturbation or data transformation to encrypt datasets.

IV. SYSTEM MODEL

In our design, we consider two major entities as shown in Fig.1: a Dataset Owner and a Cloud Server. The owner has a collection of data objects, which will be outsourced to the cloud server for clustering after encryption. The cloud server performs the K-means clustering directly over the encrypted dataset without any decryption. During the clustering, the cloud server interacts with the owner for a small amount of encrypted intermediate inputs/outputs. The clustering is finished when the clustering results do not change any more, or a predefined number of iterations are reached. As shown in figure 1.

4.1 MapReduce Framework

MapReduce is the programming framework for processing large scale datasets in a distributed manner. This task process with massive amounts of data, MapReduce divides the task into two phases: map and reduce. These two phases are expressed with map and reduce functions, which take <key, value> pairs as input and output data format. In a cluster, nodes that are responsible for map and reduce functions are called mappers and reducers respectively.

In a MapReduce task, the framework splits input datasets into data chunks, which are processed by independent mappers in parallel.

Each map function processes data and generates intermediate output as <key, value> pairs. These intermediate outputs are forwarded to reducers after shuffle. According to the key space of <key, value> pairs in intermediate outputs, each reducer will be assigned with a partition of pairs. In MapReduce, intermediate <key, value> outputs with the same key are sent to the same reducer. After that, reducers sort and group all intermediate outputs in parallel to generate the final results.

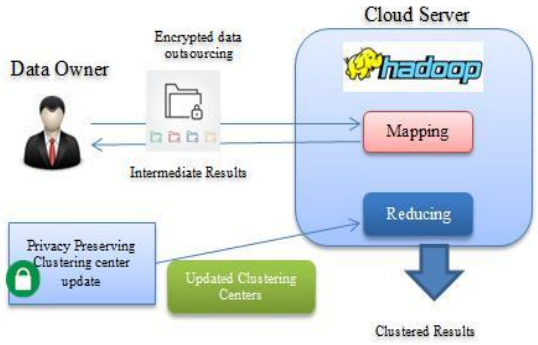

Figure 1 Proposed system Architecture

V. CONSTRUCTION OF NOVEL SECURED MAPREDUCE BASED K-MEANS CLUSTERING

Our scheme consists of three stages as shown in above figure:

1) System Setup and Data Encryption;
2) Single Round MapReduce Based Privacy-preserving Clustering
3) Privacy preserving Clustering Center Update.

In Stage 1, the owner first setups the system by selecting parameters for K-Means and MapReduce. The owner then generates encryption keys for the system, and encrypts the dataset for clustering.

In Stage 2, the cloud server performs a round of clustering and allocates encrypted objects to their closet clustering centers. After that, the cloud server returns a small amount of encrypted information back to the owner as the intermediate outputs.

In Stage 3, the owner updates clustering centers based on information from the cloud server and his/her own secret keys. These new centers are sent to the cloud server in encrypted format for the next round of clustering. Stage 2 and Stage 3 will be iteratively executed until the clustering result does not change any more or the predefined number of iterations is reached.

5.1 K-means Clustering Algorithm

K-means clustering algorithm aims at reallocate a set of data objects into k disjoint clusters, each of which has a center. For each data object, it will be assigned to the cluster whose center has shortest distance to the object. Data objects and centers can be denoted as multi-dimensional vectors, and their distances can be measured using the square of Euclidean distance.

K-means clustering is an iterative processing. The algorithm selects k initial cluster centers, and all data objects are allocated into the cluster whose center has the shortest distance to them. For each data object, it will be assigned to the cluster whose center has shortest distance to the object. Data objects and centers can be denoted as multi-dimensional vectors, and their distances can be measured using the square of Euclidean distance. After a round of clustering, centers of clusters are updated. Particularly, the new center of a cluster is generated by averaging each element over all data object vectors in the same cluster.

5.2 Proposed System: Flow diagram

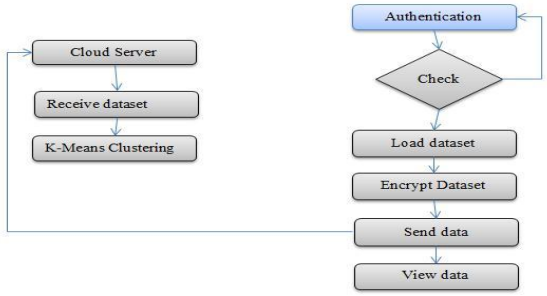

Figure 2. Proposed Flow chart

5.3 User Login Process

User can login the system with credentials such as user name and password. Once user enter in to the system and it check with the sever, if it valid then system move to load dataset methods. The dataset describes with set of attributes such as the product id, product url, product stock, product price and discount price.. Then we find the accuracy and scalability over K-means clustering algorithm, after loading our dataset, we encrypt the dataset for providing privacy and security, in this regards we used Homomorphic encryption after that encrypted data will be pass on to the cloud server.

5.4 Data Owner

After send the cloud the server, the cloud server only has to all the encrypted data objects in the datasets all the encrypted clustering centers and all intermediate output landed in cloud center. Based on the intrusion key we further K-means clustering process in the privacy preserving manner. In which cloud server only have accessed encrypted data without any decryption

5.5 Privacy preserving K-means

K-means clustering algorithm aims at reallocate a set of data objects into k disjoint clusters, each of which has a center. For each data object, it will be assigned to the cluster whose center has shortest distance to the object. Data objects and centers can be denoted as multi-dimensional vectors, and their distances can be measured using the square of Euclidean distance. K-means clustering is an iterative processing. The algorithm selects k initial cluster centers, and all data objects are allocated into the cluster whose center has the shortest distance to them. After a round of clustering, centers of clusters are updated. Particularly

Algorithm 1: K-means Clustering

Input: k: number of clusters; max: a predefined

number of iterations; n data objects

$\vec{D}_i$ □□□$d_{i1}, \cdots, d_{im}$□□,1□□□i□□□□n

Output: k clusters **begin**

Select k initial cluster centers $\vec{C}_x$, 1 □ x □ k **While** *max* □ 0 **do**

1. Assign each data object ~Di to the cluster center with minimum distance Dist($\vec{D}_i$; ~ $\vec{C}_x$)to it.

2. Update $\vec{C}_x$ to the average value of those $\vec{D}_i$

Assigned to the cluster x. Output k reallocated clusters.

VI. PERFORMANCE ANALYSIS

To evaluate the performance of our privacy-preserving MapReduce based K-means clustering scheme in terms of efficiency, scalability, and accuracy, we implemented a prototype on different domains of datasets. To demonstrate that our scheme introduces reasonable computation and communication overhead for privacy guarantee, we also implemented a non-privacy-preserving Map Reduce based K-means under the same configuration. All experimental results represent the mean of 10 trials.

Efficiency: In our evaluation, we focus on evaluating the efficiency of a single round clustering, because different rounds of clustering have the same computational cost on the owner and

the cloud server. In addition, the number of clustering rounds is mainly determined by the dataset itself and the selection of initial clustering centers, and is independent to the design our scheme.

VII. CONCLUSION

In this paper we proposed a novel privacy-preserving MapReduce based K-means clustering scheme in big data framework, this scheme achieves clustering speed and accuracy that are comparable to the K-means clustering without privacy protection. We provide thorough analysis to show the security and efficiency of our scheme. Make it extremely suitable for parallelized processing in cloud computing environment.

REFERENCE

[1] J. Vaidya and C. Clifton, "Privacy-preserving k-means clustering over vertically partitioned data," in ACM SIGKDD, 2003, pp. 206–215.
[2] C. Su, J. Zhou, F. Bao, T. Takagi, and K. Sakurai, "Two-party privacypreserving agglomerative document clustering," in ISPEC. SpringerVerlag, 2007, pp. 193 – 208.
[3] G. Jagannathan and R. Wright, "Privacy-preserving distributed k-means clustering over arbitrarily partitioned data," in ACM SIGKDD, 2005, pp. 593–599.
[4] P. Bunn and R. Ostrovsky, "Secure two-party k-means clustering," in ACM CCS, 2007, pp. 486–497.
[5] C. Gentry, "Fully homomorphic encryption using ideal lattices," in ACM STOC, 2009, pp. 169–178.
[6] C. Gentry and S. Halevi, "Implementing gentry's fully-homomorphicencryptionscheme,"in EUROCRYPT. Springer, 2011, pp. 129–148.
[7] R. Agrawal and R. Srikant, "Privacy preserving data mining," in ACM SIGMOD, vol. 29, 2000, pp. 439–450.
[8] Y. Lindell and B. Pinkas, "Privacy preserving data mining," in Journal of Cryptology, vol. 15, 2002, pp. 177 – 206.

