International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Two-Phase Privacy Preserving Big Data Hybrid Clustering for Multi-Party Data Sharing

[1]Manjula GS, [2]Dr. T. Meyyappan

Submitted: 28/04/2023 **Revised**: 27/06/2023 **Accepted**: 06/07/2023

Abstract: In this study, we take on the challenge of private data clustering. Performing a clustering technique on the union of datasets held by many parties without disclosing any further information is a scenario that has been investigated. This issue, an instance of protected multi-party computing, may be addressed using existing protocols. DBSCAN and K-Medoid apply to all data types and produce clusters identical to conventional ones, while other clustering methods are only relevant to certain kinds of data. As its name implies, DBSCAN and K-Medoid are algorithms best suited for use with a single database. In this study, we propose a method for determining the separation of data points when the information is split across two servers. This study proposes a novel method, a modified version of the privacy-preserving hybrid clustering algorithm that may be used on data sets that have been vertically and horizontally partitioned and are spread over numerous nodes in a network. The results of the experiments showed that the new technique outperformed the old ones.

Keywords*: Big data, clustering, multiparty communication, partitioned database, privacy-preserving, secure channel*

1. Introduction

Various sources, such as social media, satellites, sensors, mobile devices, computer simulations, and business transactions, contribute to the daily deluge of data [1]. Insightful information may be gleaned from this data and used in various fields, including business intelligence, forecasting, decision support, and in-depth analysis. Because of the sheer volume of data, data mining is essential for obtaining the desired insights [2, 3]. Big Data (BD) refers to very large data sets; for example, Facebook has over 30 petabytes of data, whereas Walmart has roughly 2.5 petabytes. Typically, there are three distinct data categories: structured, semi-structured, and unstructured. Most enterprises have many unstructured documents in reports, emails, or web pages. Within the next two years, the size of this data is predicted to exceed 44 zettabytes [4]. Cloud providers provide scalable and affordable services for storing, processing, and analyzing huge amounts of data, commonly known as big data. As a result, corporations adopt cloud services to alleviate the load of keeping big amounts of data locally. However, outsourced contents (documents) frequently include private or sensitive information such as health records of patients, financial reports of enterprises, or criminal records of police departments [5] that must be safeguarded from inside and outside cloud adversaries. Many businesses are still reluctant to move their operations to the cloud because of security and privacy concerns. Specifically, the terms "big data" and "large data sets" refer to three different characteristics: volume, velocity, and diversity [6]. The pace at which information is created, gathered, and transmitted is known as "the velocity of data." Data may be accessed in various places, but not always in a way that facilitates simple comprehension [7]. The availability of such large datasets and the processing capacity of the cloud have stoked considerable interest in machine learning (ML). In supervised ML methods, such as neural networks, the known data records are used to train a model that is subsequently used for additional tasks, such as classifying the unknown data.

Conversely, unsupervised ML techniques don't rely on a "training" phase and instead aim to unearth hidden patterns and structures in raw, unlabeled data. Clustering is an unsupervised ML method for discovering data groups with similar characteristics. Clustering algorithms have a broad variety of privacy-sensitive applications, including financial analytics, market research, and medical diagnostics, all of which need the protection of sensitive commercial or personal data. [8, 9, 10]. In most cases, the quality of the findings increases as more data is clustered from more sources. However, it requires more computation and storage capacity that has to be hired from third-party sources like cloud servers. When doing this, the confidential data must be safeguarded from untrustworthy servers and other data owners. As the

1 Research Scholar, Department of Computer Science, Alagappa University, Karaikudi
2 Professor, Department of Computer Science, Alagappa University, Karaikudi,
Tamil Nadu, India
E-mail: 5gsmanjula@gmail.com

amount of available data has grown steadily over the last several years, many researchers have turned their attention to figuring out how to effectively assess and utilize massive amounts of raw data to draw meaningful conclusions [11]. Data mining relies heavily on cluster analysis because it can examine the similarities between data points and the underlying structure of massive amounts of raw data without requiring prior knowledge [12]. Clustering separates similar components into a single group while identifying and grouping elements with distinct qualities.

Cluster analysis has several practical applications [13, 14, 15, and 16]; some examples include internet search, picture and face recognition, and cyber security. Clustering algorithms encounter new challenges as the amount of data rises and new forms of data emerge. Clustering algorithms must be able to analyze multiple forms or types of data sets, be scalable, manage noise, and eliminate human involvement [17, 18]. Clustering can be done based on density, grid, hierarchy, or model. These algorithms employ various processing methodologies in response to data sets [19, 20, and 21]. The major challenges of clustering algorithms are estimating the value of clusters, perfection in clustering, processing efficiency, privacy protection, etc. [22]. Many obstacles stand in the way of protecting people's privacy while working with massive amounts of data (BD). Data consumers, data collectors, data miners, and decision-makers are only a few of the positions whose privacy is at risk, as discussed by the authors of [23]. It's possible to take a few steps to safeguard the confidentiality of information utilized in these capacities. In this work, we have thought about the role of the data miner and the problem of protecting individual privacy while using data mining methods. One method offered to protect sensitive information during the development of a naive Bayesian (NB) classification may be found in [24]. Clustering techniques that don't leak private data are described in [25, 26]. However, this field faces some obstacles. Most clustering algorithms used in distributed BD systems aren't secure, and only a select handful, like k-means, has been tweaked to ensure user anonymity. This work introduces a refined version of Density-Based Spatial Clustering of Applications with Noise (DBSCAN) that may be employed in openly available big data platforms without endangering individual privacy.

The paper's main contribution is multi-party data privacy hybrid clustering in big data. Hybrid clustering has invoked the DBSCAN, and the K-medoid clustering algorithm has named as mpartDBSCAN method.

Similar to k-means and medoid-move, the k-medoids method does clustering. Partitioning (separating the data set into distinct categories) is performed using the k-means and k-medoids algorithms. To cluster a collection of n items into a fixed number of clusters, K-medoid uses the tried-and-true approach of k-means clustering. The plan is a useful tool for solving k. When k-means results looked to be different, it was more forgiving of commotion and unusual circumstances.

The following is the plan for this paper: The literature on clustering and protecting individual privacy in a large dataset is discussed in Section 2. Section 3 introduces the reader to the DBSCAN algorithm's fundamental ideas and provides a detailed description of the proposed process with hybrid clustering. Section 4 details the experiments conducted, while Section 5 provides an overview and analysis.

2. Literature Survey

Different strategies for guaranteeing the privacy of personally identifiable information, as well as their benefits and drawbacks, were discussed in this section. It's no secret that privacy-friendly data mining is all the rage in academia now. To protect users' anonymity, researchers Rakesh Agrawal and Ramakrishnan Srikanth have suggested privacy-protecting data mining [27]. Jha et al. [28] presented a distributed privacy-protective solution for k-means clustering for horizontally partitioned data. Vaidya and Clifton [29] highlighted the privacy-protecting clustering that happens when data was partitioned vertically. Prasad and Rangan presented BIRCH, a method for protecting individual privacy while dealing with vertically partitioned huge datasets. [30]. Jagannathan and Wright [31] suggested a k-means clustering method that protects user anonymity and created databases that may be arbitrarily divided. Recently, several experts have been studying privacy-preserving machine learning (PPML) [32, 33, 34, 35]. Several polls have been taken to comprehend the constantly developing scientific subject better. Haralampieva et al. [36] investigated present systems in private image classification. [37] Provides a high-level review of frameworks for private neural network inference. Many studies [38, 39] have looked at the issues with confidentiality that accompanies statistical databases. Two primary methods may be used to ensure their safety when developing machine learning algorithms. The first method transforms the data set to add noise before using the algorithm. Several researchers [40, 41, 42] have used this approach in their efforts to design anonymous clustering algorithms. Taking inspiration from the secure-multiparty computing literature was another method for creating privacy-protecting algorithms. This strategy was preferred over

perturbation since formal guarantees of privacy were provided for these algorithms. The second method was used in this article. Bunn and Ostrovsky [43] construct an AHE-based S2PC-based two-party private K-means protocol for arbitrary data partitioning. On the other hand, using costly AHE results in implausible performance estimates.

Jäschke et al. [44] use fully homomorphic encryption (FHE) to ensure privacy while outsourcing K-means. They reduce the original K-means clustering's runtime cost by approximating the centroid discovery and distance comparisons. However, the cost was still too high for large data sets to justify the effort. Improved data privacy was the focus of [45], where the authors introduce the Fuzzy based cell generalization approach. Mohassel et al. [46], working within the semi-honest security paradigm, provide a privacy-preserving Kmeans technique that uses Yao's garbled circuits. K-means allows them to work more efficiently since it only needs a single, constant input, a point in the input data, to compute the same distance function to all (repeatedly updated) centroids. Rahman et al. [47] propose a key-HE2-based DBSCAN protocol for outsourcing scenarios in which data owners cooperate with an untrusted server to perform clustering operations while protecting their data's confidentiality. Unfortunately, it sends information to the server, such as cluster sizes and neighborhood patterns. That work did not include any implementation or performance evaluation. In [48], the authors have proposed a privacy-preserving technique using association rule mining for sensitive items in market-based databases. Anonymization technique through record elimination technique has been proposed in [49].

3. Classic Dbscan Clustering Algorithm

Ester et al. in [50] invented DBSCAN to address the shortcomings of well-established clustering techniques like the K-means algorithm. DBSCAN can identify clusters of any kind. In addition, the data-driven clustering method allows for the number of clusters to be modified as needed. The system treats outliers as noise and hence disregards them. DBSCAN outperforms K-means on four different data sets, as seen in Fig. 1.

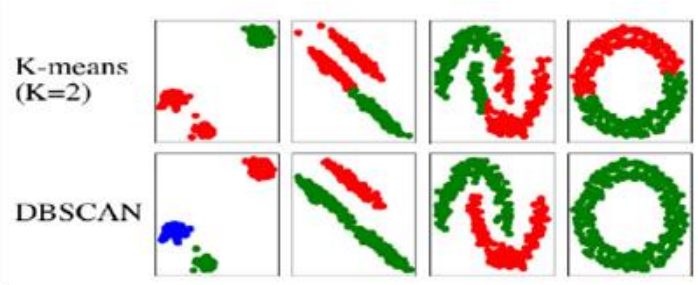

Fig.1 Cluster formation of K-means and DBSCAN algorithms for four different data sets

The DBSCAN algorithm is built on the concept that each cluster has a high point density. The steps in the classic DBSCAN algorithm are as follows:

1. set *Epsilon*- a dense region's neighborhood and find *N-epsln* the least number of points necessary to construct the dense region.

2. Choose an element that hasn't been visited yet (*first_elmnt*), mark it as visited, then count the number of elements in Eps (*nmbr_elmnt*).

3. If *nmbr_elmnt* is less than *N-epsln*, treat *first_elmnt* as noise; otherwise, add *first_elmnt* to the cluster and add elements from *Epsilon* to the array.

4. repeat steps 2-3 for all the elements in the array

5. Show the cluster after all array elements are considered;

3.1 Privacy-Preserving Multi-Party Hybrid Clustering (mpartDBSCAN) Algorithm

Data sources in the distributed data mining paradigm are likely to be located in more than one physical place. Vertical, horizontal, or random partitions might separate the information. Due to vertical data partitioning, the same entities are being tracked at many locations but with different feature sets. In horizontal data partitioning, the same data is collected by many sources but for different users.

3.2 Problem Statement

Let us assume that the database *DB* contains n records. Each record R_i consists of *m* attributes (fields/columns). For understanding, it is assumed that data is distributed among two users Pand Q. The portion of the database held by the user P is represented by DB_P and the portion of the database held by user Q is represented by DB_Q such that $DB = DB_P \cup DB_Q$. The clustering algorithm tries to form a cluster on DB_P and DBQ where Phas no knowledge of *DBQ* and Q does not know DB_P.

Now the vertically and horizontally partitioned DB_P and $DB_Q$are defined as follows:

Definition 1. Vertically partitioned database: P'spartition *is represented by* DB_P= {Rp_1,Rp_2,..Rp_n} and the Q's partition is represented by DB_Q={Rq_1, Rq_2,...Rq_n}. For every record Ri, P has values of 1 to k attributes, and Q has k+1 to n attributes. Therefore, the actual record Ri= DB_P + DB_Q.

Definition 2. Horizontally partitioned database:P'spartition *is represented by* DB_P= {Rp_1,Rp_2,..Rp_m,} contains 1 to m records of the database with n attributes, and the Q's partition is represented by DB_Q={Rq_{m+1}, Rq $_{m+2}$,...Rq_{m+n}}. Therefore, the actual record DB= DB_PU DB_Q.

3.3 Proposed Multiparty, Privacy-Preserving DBSCAN, and K-medoid Algorithm

The proposed approach has been renamed mpartDBSCAN, as its major goal is to protect sensitive information at all data processing and transmission phases in distributed systems using network channels. The algorithm uses the notations shown below:

- *Epsilon*– epsilon region (Area of neighbor elements);
- *N-epsln*– smallest number of elements required to construct*Epsilon;*
- *n* – Total number of elements considered;
- *r* – The total amount of participants;
- *first_elmnt*–the element under consideration;
- *nmbr_elmnt(a)* – value of elements in the array a[] (value of elements within *Epsilon*);
- *clust_id*– cluster identifier (value used to count the cluster);
- *clust [clust_id]* – Array of cluster;
- *a[]* –Array which contains the elements around *first_elmnt*
- *b[]* – Array with the elements which are located around*a[]* array elements.

The algorithm below shows the steps followed by the privacy-preserving K-medoid and DBSCAN clustering method.

This algorithm considers each unvisited objects *first_elmnt* and calculates the value of points *nmbr_elmnt* inside *Epsilon*. If *nmbr_elmnt* is greater than or equal to *N-epsln*, item *first_elmnt* is added to the cluster, and objects inside *Epsilon* are added to a[] array for subsequent consideration. The algorithm additionally employs a second array b[], which contains items that are included inside the *Epsilon* of the [] array elements.

Algorithm I //privacy-preserving mpartDBSCAN

```
forfirst_elmnt= 1 to ndo
{
  if the elementfirst_elmnt is not visited
  {
    element first_elmnt is marked as visited
    calculate nmbr_elmnt// Call Algorithm II
    ifN-epsln<= nmbr_elmnt(a[])
    {
clust_id = clust_id + 1
        cluster [clust_id] = first_elmnt
        Analyze a[]array
    }
  }
}
```

In a cloud environment that handles big data of multiple clients, privacy-preserving between clients is an essential functionality of distributed system nodes. To maintain privacy, every node has to calculate distance vector:

$$Xi = first\_elmnt - second\_elmnt \quad (1)$$

The first elmnt and second elmnt element distances are shown in equation (1). After that, they settle on a value for nmbr elmnt. If this sensitive information is sent through an unsecured route, an invasion of privacy might result. The data will be processed at nodeNr. This node is responsible for implementing the cluster's functionality and collecting data from other nodes. The most effective method of keeping personal information secret is encrypting the connection between Nr and other nodes. However, this is not enough for data mining methods since Nr will have access to the information of the other participants in the study, resulting in the leak of such information. Therefore, data privacy for external

adversaries and other data storage locations must be provided in this situation.

3.4 K-Medoid Clustering Algorithm

K-Medoid is another approach for grouping data that uses partitions. Clusters are shown as medoids. The term "medoid" refers to the data part of a collection whose average dissimilarity to all the former member of the set is low, and so indicates the most centrally situated data item of the set.

Input: value of clusters k, the data set containing n items D

Output: A set of k clusters that minimize the Sum of the objects' dissimilarities to their nearest medoids.

$$z = \sum_{i=1}^{k} \sum |x - mi|$$

Where Z is the Sum of ABE for all items in the data set, x is the data point in the space expressive a data item, and m_i is the medoid of cluster C_i

1. Pick k starting medoids from the data set at random.

2. find the medoid closest to each remaining data point and put it in that cluster.

3. Choose a non-medoid data item at random and calculate the total cost of replacing the original medoid data item with the new one.

4. The swap operation is carried out to produce the new set of k-medoids if its total cost is less than zero.

5. After the medoids have settled into their new positions, repeat steps 2–4.

A channel to be established where all nodes can submit their data to Nr. This is done by using homomorphic encryption. The following method will be used to ensure data privacy. As the first step, a node N_1 for transmitting its data is chosen. N_1 will protect the privacy of its data before transmitting it to Nr. Similarly, all other nodes associated with Nr will encrypt their data and submit it to Nr. Transmitting data to Nr will result in the disclosure of their personal information. So, for homomorphic encryption, each node "i" produces its key pair (Ei, Di), where Ei is the public key used for encryption and Di is the secret key used for decrypting.

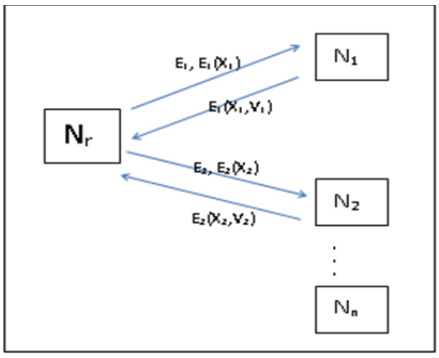

Fig 2. Pattern of communication between N_r and other nodes

Algorithm II // Clustering of elements

{
saved_elmnt=scnd_elmnt
forscnd_elmnt = 1 to n**do**
{
fori = 1 to r **do**
Ni :$\overrightarrow{Xi}$ =$|first\_elmnt - scnd\_elmnt|$ //Node Ni calculates the vector $\overrightarrow{Xi}$
}
Nr creates the vector $\overrightarrow{Vi}$:$\sum_{i=1}^{r} \overrightarrow{Vi}$ for every participating node Ni.
fori = 2 to r**do**
{
Ni creates the key pair (Ei,Di)
Ni sends (E,($\overrightarrow{Xi}$),Ei) to Nr
Nr sends back the altered data (Ei($\overrightarrow{Xi} + \overrightarrow{Vi}$))
Ni computes vector $\overrightarrow{Xi'}$ = Di(Ei($\overrightarrow{Xi} + \overrightarrow{Vi}$))
}
} //End of the Algorithm II

Node Nr receives an encrypted vector Ei = $(\overrightarrow{Xi})$ and the encryption key Ei during privacy-preserving clustering. Nr creates a random vector $(\overrightarrow{Vi})$[51] and encrypts it using Ei. Then it computes the privacy-preserving vector for node Ni by using the equation (2) as follows:

$$Ei(\overrightarrow{Xi})+Ei(\overrightarrow{Vi}) = Ei(\overrightarrow{Xi}+\overrightarrow{Vi}) \quad (2)$$

Nr delivers this vector to the corresponding node that uses Di's secret key to decrypt it. As a result, Nr is unaware of the vectors of other parties, and the keys of other parties are unknown to those who get updated vectors.

4. Experimentation Results

The system with the following configurations, as shown in table I, is used:

Table I Experiment setup

Processor	Intel Core CPU@2.90Ghz
Memory	4.0 GB
Operating System	Windows 10
Analyzer	MATLAB R2013a

We have employed the widely-used D31, t1.2k, t5.8k, and t7.10khas databases to test and compare the effectiveness of our approach [52]. Two of them, t5.8k and t7.10k, are unlabeled chameleon datasets [53], whereas the others are collected in the usual way from [54]. These databases should be chosen because the original DBSCAN and K-medoid clustering techniques use these data sets, and (i) these databases allow for repeated iterations of the studies. Table 2 displays the primary features of the datasets. The table below shows the value of elements (n), features (d), and clusters (k) for a variety of commonly used datasets.

Table 2. Characteristics of the datasets used in this research:

Sl.no	Name of the Dataset	n	d	k
1	R15	600	2	15
2	D31	3100	2	31
3	t4.8k	8000	2	6
4	t7.10k	8000	2	8

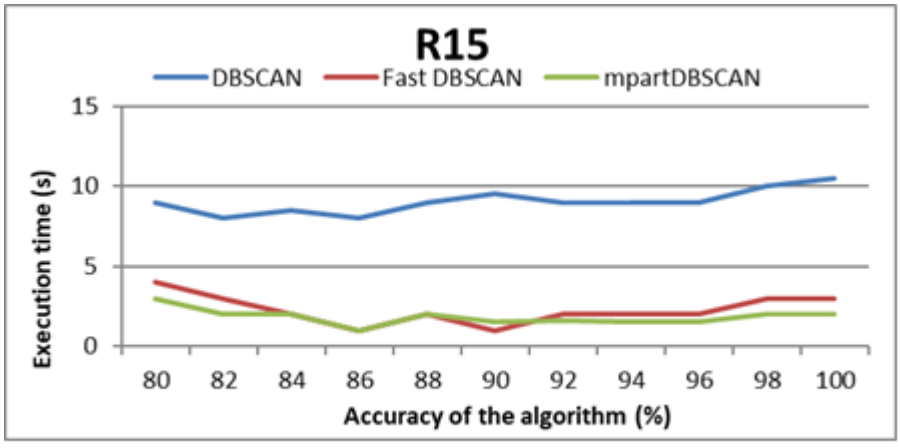

Fig. 3. Comparison of accuracy of execution time for R15 dataset

In the figures from Fig.3 to Fig. 6, the algorithm's accuracy and the execution time of DBSCAN, FastDBSCAN[55], and the proposed mpartDBSCAN for various data sets has been compared.

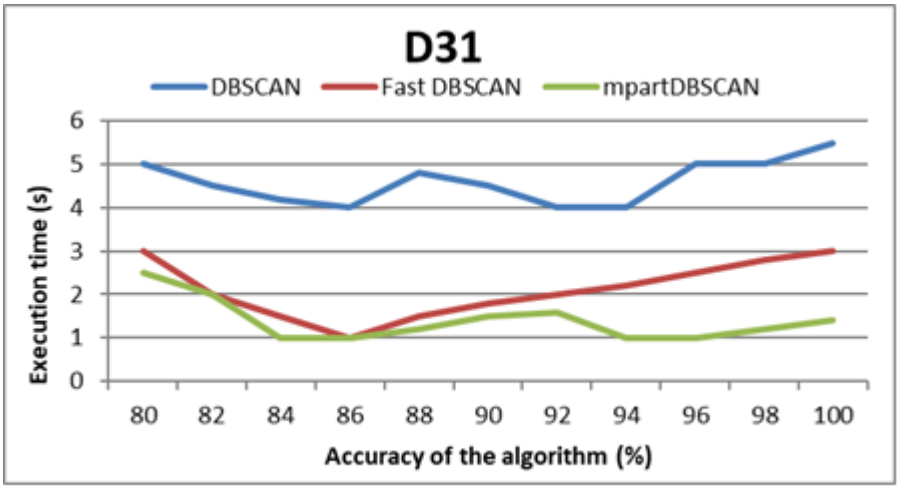

Fig. 4. Evaluation of accuracy of execution time for D31 dataset

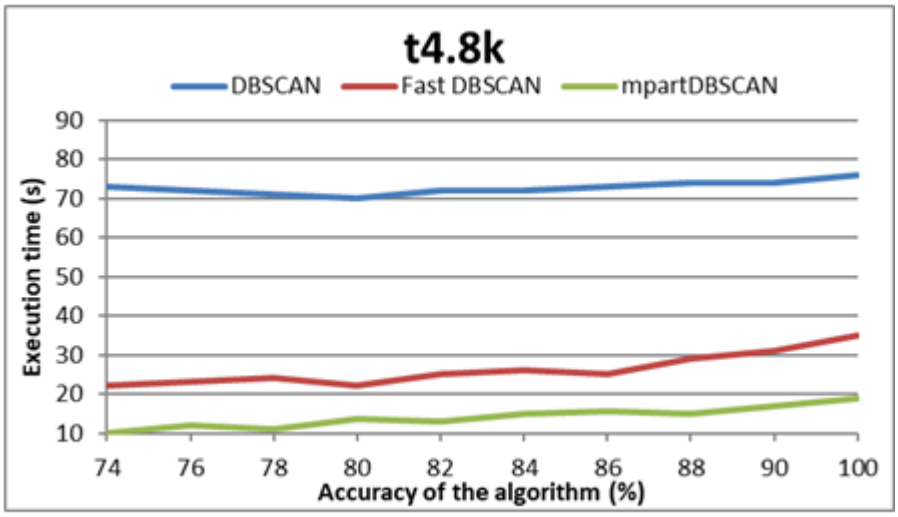

Fig. 5. Evaluation of accuracy of execution time for t4.8k dataset

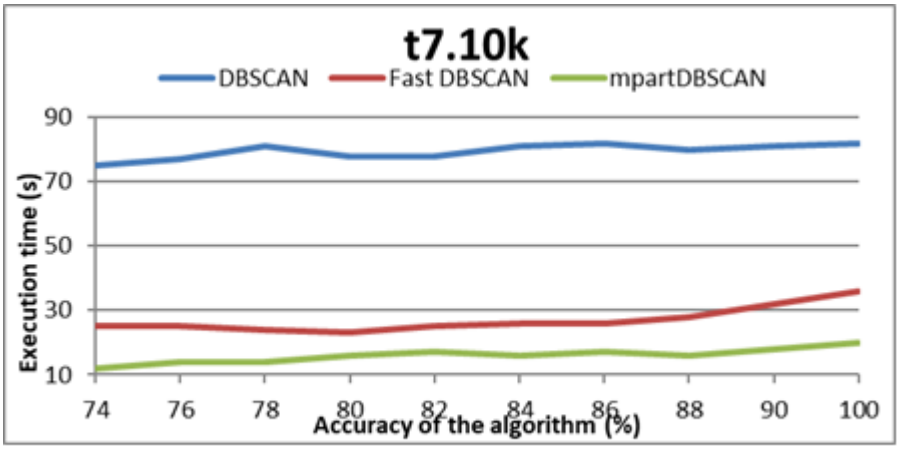

Fig. 6. Evaluation of accuracy of execution time for t7.10k dataset

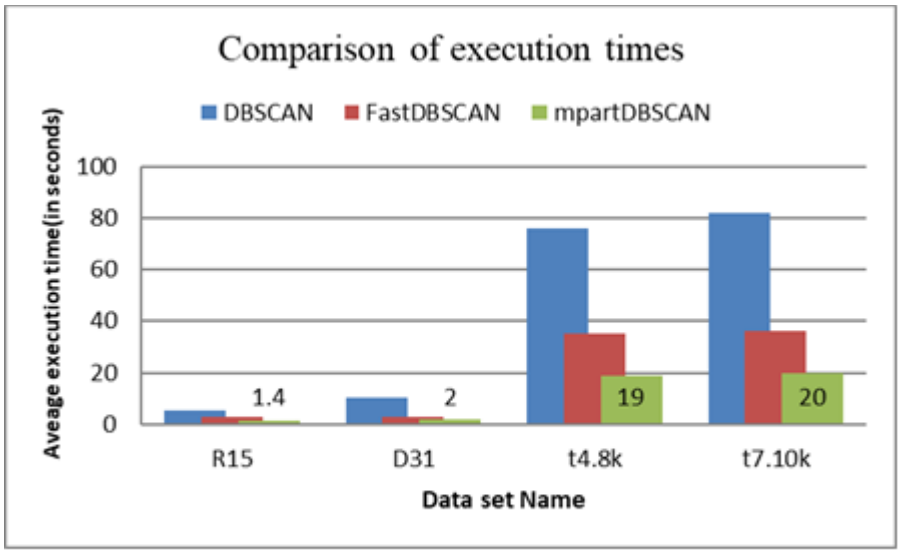

Fig. 7. Evaluation of execution time for different data sets

5. Conclusion

This study proposes mpartDBSCAN that utilizes secure multi-party processing. DBSCAN is universally applicable to any data type, unlike most well-known privacy safeguards. DBSCAN and K-medoid create clusters that look like those seen in nature. This research suggested a safe distance metrics method for vertically or horizontally partitioned databases. Based on these protocols hybrid clustering method for distributed data is outlined. How this protocol may be used for negative databases and databases with arbitrary partitions will be looked at in the future.

References

[1] Chen, C.L.P., Zhang, C.Y. (2014), "Data-intensive applications, challenges, techniques and technologies: A survey on Big Data." Inf. Sci. (Ny)., 275: 314-347. https://doi.org/10.1016/j.ins.2014.01.015

[2] Wang, X.K., Yang, L.T., Liu, H.Z., Deen, M.J. (2017). "A big data-as-a-service framework: State-of-the-art and perspectives." IEEE Trans. Big Data, 4(3): 325-340. https://doi.org/10.1109/TBDATA.2017.2757942

[3] Elkano, M., Sanz, J.A.A., Barrenechea, E., Bustince, H., Galar, M. (2019). CFM-BD: "A distributed rule induction algorithm for building Compact fuzzy models in big data classification problems". IEEE Trans. Fuzzy Syst., vol. 1. https://doi.org/10.1109/TFUZZ.2019.2900856

[4] www.entrepreneur.com/article/273561. Accessed Nov. 18

[5] SmZobaed and Mohsen Amini Salehi. "Big data in the cloud. In Encyclopedia of Big Data", Edited by Albert Zomaya and SherifSakr, Springer International Publishing.

[6] Kumar, D., Bezdek, J.C., Palaniswami, M., Rajasegarar, S., Leckie, C., Havens, T.C. (2016). "A hybrid approach to clustering in big data. IEEE Transactions on Cybernetics", 46(10): 2372-2385. https://doi.org/10.1109/TCYB.2015.247741

[7] Rayala, Venkat, and Satyanarayan Reddy Kalli. "Big Data Clustering Using Improvised Fuzzy C-Means Clustering." *Rev. d'IntelligenceArtif.* 34, no. 6 (2020): 701-708.

[8] M. Ahmed, A. N. Mahmood, and Md. R. Islam. 2016. "A Survey of Anomaly Detection Techniques in Financial Domain. In Future Generation Computer Systems".

[9] Q. Guo, X. Lu, Y. Gao, J. Zhang, B. Yan, D. Su, A. Song, X. Zhao, and G.Wang. 2017. "Cluster Analysis: A New Approach for Identification of Underlying Risk Factors for Coronary Artery Disease in Essential Hypertensive Patients". In Scientific Reports.

[10] G. Punj and D.W. Stewart. 1983. "Cluster Analysis in Marketing Research: Review and Suggestions for Application", In Journal of Marketing Research.

[11] J. Hou, W. Liu, "Evaluating the density parameter in density peak based clustering", 2016, pp. 68–72.

[12] Y. Wang, D. Wang, X. Zhang, W. Pang, C. Miao, A. Tan, Y. Zhou, Mcdpc: "multi-center density peak clustering", Neural Comput. Appl. (2020) 1–14.

[13] Y. Shi, Z. Chen, Z. Qi, F. Meng, L. Cui, "A novel clustering-based image segmentation via density peaks algorithm with mid-level feature", Neural Comput.Appl. 28 (2017) 29–39.

[14] A.D. Marco, R. Navigli, "Clustering and diversifying web search results with graph-based

word sense induction", Comput. Linguist. 39 (2013) 709–754.

[15] Z. Du, "Energy analysis of internet of things data mining algorithm for smart green communication networks", Comput. Commun. 152 (2020) 223–231.

[16] S. Aghabozorgi, Y.W. Teh, "Stock market co-movement assessment using a three-phase clustering method", Expert Syst. Appl. 41 (2014) 1301–1314.

[17] L. Sun, G. Chen, H. Xiong, C. Guo, "Cluster analysis in data-driven management and decisions", J. Manag. Sci. Eng. 2 (2017) 227–251.

[18] J. Zhou, Z. Lai, C. Gao, X. Yue, W.K. Wong, "Rough-fuzzy clustering based on two-stage three-way approximations", IEEE Access 6 (2018) 27541–27554.

[19] T. Gocken, M. Yaktubay, "Comparison of different clustering algorithms via genetic algorithm for vrptw", Int. J. Simul. Model. 18 (2019) 574–585.

[20] K.M. Kumar, A.R.M. Reddy, "An efficient k-means clustering filtering algorithm using density based initial cluster centers", Inform. Sci. 418 (2017) 286–301.

[21] R.C. Hrosik, E. Tuba, E. Dolicanin, R. Jovanovic, M. Tuba, "Brain image segmentation based on firefly algorithm combined with k-means clustering", Stud. Inf. Control 28 (2019) 167–176.

[22] S. Sieranoja, P. Franti, "Fast and general density peaks clustering", Pattern Recognit. Lett. 128 (2019) 551–558.

[23] Lei Xu, Chunxiao Jiang, Jian Wang, Jian Yuan, Yong Ren. "Information Security in Big Data: Privacy and Data Mining", IEEE Access, 2014, pp. 1149-1176

[24] J. Vaidya, M. Kantarcoglu, and C. Clifton, "Privacy-preserving Naïve Bayes classification," Int. J. Very Large Data Bases, vol. 17, no. 4, pp. 879–898, 2008.

[25] R. Akhter, R. J. Chowdhury, K. Emura, T. Islam, M. S. Rahman, and N. Rubaiyat, "Privacy-preserving two-party *k*-means clustering in malicious model", in *Proc. IEEE 37th Annu. Comput. Softw. Appl. Conf. Workshops (COMPSACW)*, Jul. 2013, pp. 121–126.

[26] I. De and A. Tripathy, "A secure two party hierarchical clustering approach for vertically partitioned data set with accuracy measure", in *Proc. 2nd Int. Symp. Recent Adv. Intell. Informat.*, 2014, pp. 153–162.

[27] 1.Agrawal, R., Srikant, R.: "Privacy preserving data mining". In: Proceedings of the 2000 ACM SIGMOD Conference on Management of Data, Dallas, TX, May 2000, pp. 439–450. ACM Press, New York (2000)

[28] Jha, S., Kruger, L., McDaniel, P.: "Privacy Preserving Clustering". In: di Vimercati, S.d.C., Syverson, P.F., Gollmann, D. (eds.) ESORICS 2005. LNCS, vol. 3679, pp. 397–417. Springer, Heidelberg (2005)

[29] Vaidya, J., Clifton, C.: "Privacy-preserving k-means clustering over vertically partitioned data". In: Proceedings of the 9th ACM SIGKDD International Conference on knowledge Discovery and Data Mining, Washington, DC, August 2003, ACM Press, New York (2003)

[30] Krishna Prasad, P., Pandu Rangan, C.: "Privacy preserving BIRCH algorithm for clustering over vertically partitioned databases". In: Jonker, W., Petkovi´c, M. (eds.) SDM 2006. LNCS, vol. 4165, pp. 84–99. Springer, Heidelberg (2006)

[31] Jagannathan, G., Wright, R.N.: "Privacy-preserving distributed k-means clustering over arbitrarily partitioned data". In: Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, Illinois, August 2005, pp. 593–599. ACM Press, New York (2005)

[32] N. Kumar, M. Rathee, N. Chandran, D. Gupta, A. Rastogi, and R. Sharma, "CrypTFlow: Secure TensorFlow inference," in *IEEE S&P*, 2020.

[33] D. Rathee, M. Rathee, N. Kumar, N. Chandran, D. Gupta, A. Rastogi, and R. Sharma, "CrypTFlow2: Practical 2-party secure inference," in *CCS*, 2020.

[34] P. Mishra, R. Lehmkuhl, A. Srinivasan, W. Zheng, and R. A. Popa, "Delphi: A cryptographic inference service for neural networks," in USENIX Security, 2020.

[35] Dhabliya, M. D. (2018). A Scientific Approach and Data Analysis of Chemicals used in Packed Juices. Forest Chemicals Review, 01–05.

[36] A. Patra, T. Schneider, A. Suresh, and H. Yalame, "ABY2. 0: Improved mixed-protocol secure two-party computation," in *USENIX Security*, 2021.

[37] V. Haralampieva, D. Rueckert, and J. Passerat-Palmbach, "A systematic comparison of encrypted machine learning solutions for image classification," in *PPMLP*, 2020.

[38] F. Boemer, R. Cammarota, D. Demmler, T. Schneider, and H. Yalame, "MP2ML: A mixed-protocol machine learning framework for private inference," in *ARES*, 2020.

[39] N.R. Adam and J.C. Wortmann. "Security-control methods for statistical databases: A comparative study". *ACM Computing Surveys*, 21, 1989.

[40] D.E. Denning. "A security model for the statistical database problem". *ACM Transactions on Database Systems (TODS)*, 5, 1980.

[41] M. Klusch, S. Lodi, and Gianluca Moro. "Distributed clustering based on sampling local density estimates". In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI 2003)*, pages 485–490, 2003.

[42] S. Merugu and J. Ghosh. "Privacy-preserving distributed clustering using generative models". In *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM 2003)*, pages 211–218, 2003.

[43] S. Oliveira and O. R. Zaiane. “Privacy preserving clustering by data transformation”. In *XVIII Simp´osioBrasileiro de Bancos de Dados, 6-8 de Outubro (SBBD 2003)*, pages 304–318, 2003.

[44] P. Bunn and R. Ostrovsky. 2007. "Secure Two-Party K-means Clustering". In CCS. ACM.

[45] Dhabliya, D. (2021a). AODV Routing Protocol Implementation: Implications for Cybersecurity. In Intelligent and Reliable Engineering Systems (pp. 144–148). CRC Press.

[46] A. Jäschke and F. Armknecht. 2018. "Unsupervised Machine Learning on Encrypted Data". In SAC. Springer.

[47] Mahesh, R., and T. Meyyappan. "Fuzzy based cell generalization to improve the data utility with minimal loss of information." *Journal of Intelligent & Fuzzy Systems* 37, no. 1 (2019): 217-225.

[48] P. Mohassel, M. Rosulek, and N. Trieu. 2020. "Practical Privacy-Preserving Kmeans Clustering". In PoPETS. Sciendo.

[49] M. S. Rahman, A. Basu, and S. Kiyomoto. 2017. "Towards Outsourced Privacy- Preserving Multiparty DBSCAN", In Pacific Rim International symposium on Dependable Computing. IEEE.

[50] Kasthuri, S., and T. Meyyappan. "Detection of sensitive items in market basket database using association rule mining for privacy preserving." In *2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering*, pp. 200-203. IEEE, 2013.

[51] Mahesh, R., and T. Meyyappan. "Anonymization technique through record elimination to preserve privacy of published data." In *2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering*, pp. 328-332. IEEE, 2013.

[52] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. 1996. "A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", In SIGKDD Conference on Knowledge Discovery and Data Mining. ACM.

[53] I.V. Anikin, K. Alnajjar, "Fuzzy stream cipher system", 2015 International Siberian Conference on Control and Communications (SIBCON), 2015. DOI: 10.1109/SIBCON.2015.7146976.

[54] Zhuo, Linlin, Kenli Li, Bo Liao, Hao Li, Xiaohui Wei, and Keqin Li. "HCFS: a density peak based clustering algorithm employing a hierarchical strategy", *IEEE Access* 7 (2019): 74612-74624.

[55] G. Karypis, E.-H. Han, and V. Kumar, "Chameleon: Hierarchical clustering using dynamic modeling," *Computer*, vol. 32, no. 8, pp. 68_75, Aug. 1999.

[56] M. Lichman, ``Uci machine learning repository," Tech. Rep. 2013.

[57] Thang, Vu Viet, D. V. Pantiukhin, and A. I. Galushkin. "A hybrid clustering algorithm: the fastDBSCAN", In *2015 International Conference on Engineering and Telecommunication (EnT)*, pp. 69-74. IEEE, 2015.

[58] Bhat , A. H. ., & H V, B. A. . (2023). E2BNAR: Energy Efficient Backup Node Assisted Routing for Wireless Sensor Networks . International Journal on Recent and Innovation Trends in Computing and Communication, 11(3s), 193–204. https://doi.org/10.17762/ijritcc.v11i3s.618

[59] Auma, G., Goldberg, R., Oliveira, A., Seo-joon, C., & Nakamura, E. Enhancing Sentiment Analysis Using Transfer Learning Techniques. Kuwait Journal of Machine Learning, 1(3). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/129