# Privacy Preservation in Big Data by Particle Swarm Optimization

Ei Nyein Chan Wai[1,2] , Aye Thida Win[1], Pei-Wei Tsai[2], Jeng-Shyang Pan[2]

[1] University of Computer Studies, Kyaing Tong, Myanmar

[2] *Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fujian University of Technology, Fuzhou, Fujian, China*

[1]einyeinchanwai@ucsy.edu.mm

*Abstract*— **We now live in the time of big data era. During these recent years, a huge amount of data consisted of text, images, audio, video and other file types is rapidly increasing and changing. When using these massive data, it is also important that these usage must not harm the privacy of data owners. That is why so many researches are worked on for this topic. Here is one of these works for privacy preservation in big data. This research constructs upon the well-known swarm intelligence technique, Particle Swarm Optimization (PSO), for clustering similar data. The novel cloud infrastructure, MapReduce Hadoop, is alsa applied to effectively handle the huge amount of so-called Big Data. Our approach is tested by using a novel UCI Adult dataset.**

*Keywords*— **Big Data, MapReduce, Privacy Preservation, HPSO.**

## I. Introduction

People are much more rely and use information and communication technology in recent decades. Because of so many data usage forms and formats, the amount of data has been exploding with un-predictable rate and coming from various sources. Trillions of bytes of data are captured to facilitate the knowledge discovery. These data are so called “Big Data”.

This massive amount of data can be used by many organizations to process and analyze the data for developing and enhancing their works. To do so, privacy is a major concern. Moreover, while trying to preserve privacy at a certain level, it is also required to hold utility on the other hand. If not, the goal of data distribution will not be achieved. To achieve the right tradeoff between privacy and utility, some researches have been proposed as a remedy of this awkward situation in recent years. Many feasible approaches are proposed, and new methods and theory come out continuously for different scenario to solve the privacy issues effectively.

In this paper, we propose a privacy preservation approach for big data based on swarm intelligence based clustering technique. Particle swarm optimization (PSO) techniques are effectively used in data clustering application. Among so many PSO branches, we apply hierarchical method in our work.

First, we use HPSO at two phases of clustering to group the data with similar attributes. Then, the resulted data groups are summarized into their general form to achieve privacy. Here, the more similar the data in same group are, the more utility they can retain. To address the scalability issue of big data, the whole approach is built upon MapReduce Hadoop infrastructure.

The remained part of this paper is organized as follows. Some preliminaries about big data and HPSO are described at Section 2. Section 3 will explain some related works that utilize MapReduce for big data privacy preservation. The detail explanation about our work can be seen at Section 4, and its experimental results are at Section 5. Section 6 will be the final section of conclusion and further implementations.

## II. Preliminaries

### A. Big Data

HACE theorem states that big data starts with large-volume, heterogeneous, autonomous sources with distributed and decentralized control, and seeks to explore complex and evolving relationships among data.

Also, IBM website of The Big Data & Analytics Hub defines the challenges of big data are discussed in respect of 5 Vs [7] as follows:

1.Volume : huge amount of data; from terabytes to exabytes.
2.Variety: limitless variety of data; text, image, video, audio, social relations, and so on.
3.Veracity : trustworthiness and authenticity of data.
4.Velocity : rapidity of data; batch or streaming.
5.Value : necessity of interdisciplinary cooperation, proportion to veracity.

### *B. Privacy Models of Big Data*

According to [9], one way of grouping the privacy models is based on the type of attack they are trying to prevent based on two categories: privacy models that counter linkage attacks and probabilistic attacks. Linkage attacks try to link one individual to a record or to a value in a given table or to establish the presence of absence in the table itself. In the case of probabilistic attacks, an attacker tries to gain as much information as possible about an individual, from a published table, beyond his own background knowledge.

Among these models, the most emphasis models for current research trends are k-anonymity and l-diversity. In privacy protection, k-anonymity model is used to prevent from record linkage attacks. A release provides k-anonymity protection if the information for each person contained in the release cannot be distinguished from at least k-1 individuals whose information also appears in the release [10]. Beyond k-anonymity model, l-diversity [11] model is proposed to guarantee privacy against attribute linkage attacks, namely homogeneity attack (positive disclosure) and background knowledge attack (negative disclosure). Homogeneity attack can achieve when all sensitive attribute values become identical for the whole block. When an advisory has background information to eliminate possible values for the sensitive attribute of an individual, background knowledge attack can occur. To attain l-diversity, the values of the sensitive attributes are well-represented in each group.

### *C. Data Anonymization for Privacy Preservation*

Data anonymization plays major role in non-interactive public data sharing process. It refers to hiding identity of sensitive data which makes sure the published data is practically useful for processing (mining) while preserving individuals' sensitive information. Normally, there is only one raw data table which includes four types of attributes, namely- identifiers, quasi-identifiers (QID), sensitive attributes (S), and non-sensitive attributes (NS). Among these, the identifier attributes are always removed when the data set is published after anonymization. Quasi-identifiers may seem harmless at first glance, but later, the sensitive data can likely be uniquely identified based only on the QID. Although anonymization is a popular approach in privacy protection, applying its traditional methods to big data can face with scalability and efficiency challenges.

Figure 1(A) is an example table of raw medical data set. In this table, Name is an identifier; Sex, Age, and Postcode are quasi-identifiers; and Illness is a sensitive attribute. After anonymization process, this table transforms to Figure 1(B).

| **Name** | **Sex** | **Age** | **Postcode** | **Illness** |
|---|---|---|---|---|
| Bill | M | 20 | 13000 | Flu |
| Ken | M | 24 | 13500 | HIV |
| Linda | F | 26 | 16500 | Fever |
| Mary | F | 28 | 16400 | HIV |

**Figure 1 (A). Raw Medical Data set**

| **Sex** | **Age** | **Postcode** | **Illness** |
|---|---|---|---|
| M | [20,24] | 13*00 | Flu |
| M | [20,24] | 13*00 | HIV |
| F | [26,28] | 16*00 | Fever |
| F | [26,28] | 16*00 | HIV |

**Figure 1 (B). Anonymous Data set of Table A**

### *D. MapReduce Model for Big Data*

To faithful the requirements of big data to support fault tolerance, parallel processing, data-distribution, load balancing, scalability and availability, Google introduced the MapReduce programming model and its open source implementation, Apache Hadoop [12].

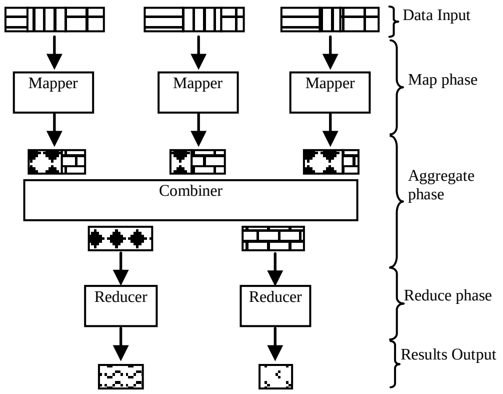

**Figure 2. MapReduce Architecture**

MapReduce consists of two different phases; Map phase and Reduce phase. A MapReduce job generally breaks the input data into chunks which are first processed by Map phase in parallel and then by Reduce phase. It works on key-value pairs (key,value). Map takes a pair (k1,v1) as input and then outputs another intermediate key-value pair (k2,v2). Reduce takes intermediate k2 and all its corresponding values list {v2} as input and outputs another pair (k3,v3) which is the intended results for users. Both Map and Reduce functions are specified by users according to their specific applications.

### *E. Hierarchical Particle Swarm Optimization (HPSO)*

PSO is an optimization technique based upon cooperation and coordination among the particles. In PSO, the swarm is initialized to a random solution set. The particles then start moving through the solution space by maintaining a velocity value V while keeping track of its best previous position achieved so far. This value is known as its personal best position (*pBest*). Global best (*gBest*) is another best value which is the best fitness achieved by any of the particles. The fitness of each particle or the whole swarm is evaluated by a fitness function.

HPSO clustering [13] combines both hierarchical clustering and partition clustering techniques and added swarm intelligence to the process to give the novel PSO based hierarchical agglomerative data clustering technique. Initially, the number of particles is kept large for the maximum coverage of the problem space. Uniform initialization helps the particles to spread in the input data space. Each particle is initialized to the data vector of the data repository using the formula:

$$loc(X(i)) = i * \left[\frac{N}{K} - 1\right] \qquad (1)$$

where loc(X) represents the location of particle in the repository, i is the index of the particle which ranges from 0 to the maximum number of particles K and N is the total number of data vectors. The Euclidean distance measure is used to find the distance between a particle and a data vector. The velocity of the particle is calculated using the standard PSO velocity update equation:

$$V_i(t+1) = w * V_i(t) + q_1 r_1 (pBest - X_i(t)) + q_2 r_2 (gBest - X_i(t)) + (Y_i(t) - X_i(t)) \qquad (2)$$

where (pBest-$X_i$(t)) is the cognitive component that controls the movement of the particle by keeping track of its best position achieved so far by that particular particle, (gBest-$X_i$(t)) is the social component that indicates the influences of other particles, and ($Y_i$(t)-$X_i$(t)) is self-organizing component takes its inspiration from the other members of that particular cluster. The new position of the particle is based on the previous position of the particle and the velocity of the particle.

The less dense clusters are merged to the nearest well populated cluster. The merging operation takes place once during each generation of the swarm. During a particular generation, a number of iterations are performed to move the particle to the most suitable position, aiming to minimize the intracluster distance. Merging of the particles is based on the average attribute values.

$$X_i = \frac{X_i(nearest) + X_i(loser)}{2} \qquad (3)$$

where $X_i$ is the newly formed particle after is merging, $X_i$(nearest) is the winner particle and $X_i$(loser) is the particle which is less populated. This approach starts from a relatively large number of particles and combining down to only one final particle. The first generation particles adjust their positions by iterating them for a particular number of iterations. The transition of swarm from one generation to another generation merges two of the selected particles and transforms the swarm into a smaller swarm.

When comparing the accuracy, HPSO has improved against hierarchical agglomerative clustering (HAC), and is also better than PSO-clustering and K-means clustering on the experiments done by [14]. Although, the efficiency has been considerably improved, it still has poor execution time as compared to partition clustering techniques.

## III.RELATED WORK

A number of researches have been done to keep this privacy intact. In this section, we summarize the recent privacy preservation works related to big data.

Zhang et al. [1] propose a highly scalable MapReduce based median-finding algorithm (MRMondrian) combining the idea of the median of medians and histogram technique. The recursion granularity is controlled to achieve cost-effectiveness by either the number of computation nodes, or recursion depth or the size of a partition. Each round of recursion involves three main steps, 1) finding the best splitting dimension and its corresponding splitting point, 2) splitting the dataset into two or multiple sub-datasets, and 3) recursively invoking such a process on the sub-datasets. This approach leverages the coefficient of variation of values of a QI-attribute to guide the selection of the best dimension. The computation of finding the median of a fixed group can be conducted in a mapper. Then, all the medians are medians are shuffled to one reducer that can find the median of medians.

A proximity privacy model with allowing semantic proximity of sensitive values and multiple sensitive attributes is proposed by Zhang et al. [2]. This model combines local recoding and proximity privacy models together to provide an anonymous dataset by means of two-phase clustering approach constructed upon MapReduce framework for scalability. The first phase of this approach, t-ancestors clustering, splits an original data set into partitions, so called β clusters that contain similar data records in terms of quasi-identifiers. Then, the proximity-aware agglomerative clustering algorithm locally recodes data partitions in parallel. The values of sensitive attributes need to be proximity aware, especially for categorical values, because if the sensitive values of the records in a QI-group of size k are identical or quite similar, adversaries can still link an individual with certain sensitive values with high confidence although the QI-group satisfies k-anonymity, resulting in privacy violation. Based on this notion of proximity for sensitive attribute values, this approach extends the proximity privacy model (ϵ,δ)k-dissimilarity to (ϵ+,δ)k-dissimilarity, where "+" implies proximity of categorical values is taken into account. Parameter k controls the size of each QI-group to prevent record linkage attacks, parameter δ specifies constraints on the number of ϵ+-neighbors that each sensitive vector can own to combat proximity attacks. A proximity-aware distance measure between two data records is defined by combining their distance and proximity index.

The use of K-means clustering for privacy preservation is proposed by Upmanyu et al. [3]. This approach uses the paradigm of secret sharing by Shatter and Merge functions upon which K-means algorithm is run. The secret have no meaningful information on their own. It can be reconstructed only when the shares are combined together. Each user computes the secret shares of their private data by means of Shatter function, and sends them over to the processing servers. The processing servers then privately collaborate to run the K-means algorithm over the secret shares without reconstructing the actual data. Chinese Remainder Theorem (CRT) is used to reconstruct the secret in Marge function.

Anonymization using Nested Clustering (ANC) for k-anonymity privacy preservation is described at [4]. This approach uses nested clustering and perturbation on each cluster by two phases architecture. During first phase, called nested clustering, the original database is clustered efficiently into enough number of sub clusters by grouping and re-clustering repeatedly. The second phase is an anonymization phase in which the numeric values are moved towards the centroid of each of the sub clusters. If the centroid of a sub cluster matches with its value, then the record is moved towards the centroid of the parent cluster. Thus the objects are made to remain in the same cluster with their values perturbed.

Lin et al. [5] proposes a well-known global heuristic search genetic algorithm (GA) based clustering approach for k-anonymization. All chromosomes of the population represent a complete solution to the problem. Each chromosome contains no fewer than k genes, where each gene indicates the index of a record in the data set. A rank-based selection strategy is adopted by sorting all possible pairs of chromosomes by the distance of the two chromosomes in each pair in ascending order, such that a higher-ranked chromosome pair (i.e., two nearby chromosomes) has a higher probability of being selected. The two chromosomes are crossed over to generate two offspring according to four crossover operations, namely random one-point, nearest-neighbor one-point, farthest member one-point, and repartition multi-point. Then, the information losses of the two offspring are calculated to determine whether they can replace their parents in the population.

A number of optimization algorithms such as particle swarm optimization (PSO), and ant colony optimization (ACO) are applied successfully in data clustering. A member of such optimization algorithms, bacterial foraging optimization (BFO), is used as clustering approach for l-diversity privacy model is expressed in [6]. This approach modified the chemotaxis step of the BFO algorithm by factorial calculus (FC) to boost the computational performance, and named as FC-BFO.

## IV. BIG DATA PRIVACY PRESERVATION BY USING HPSO

The proposed approach consists of two phases. In the first phase, a MapReduce job is done to produce the predefined numbers of intermediate β clusters. Next, a MapReduce job of HPSO lustering is executed on each β cluster. The final resulted small clusters of phase 2 are the groups of data that possess similar quasi-identifiers. Thus, they are ready to generalize to transform into their anonymized forms.

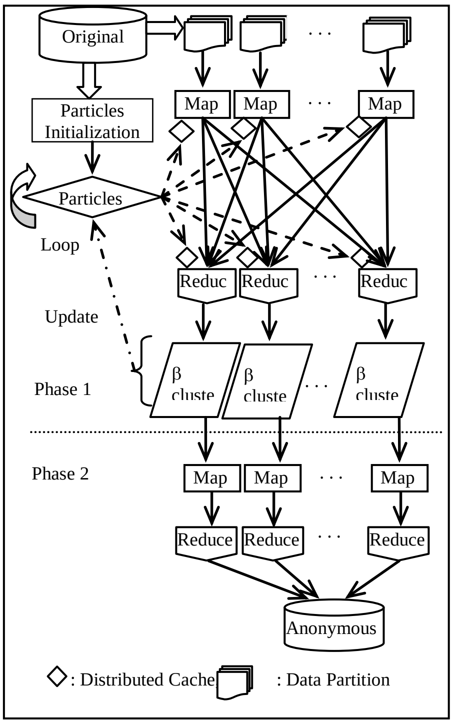

**Figure 3. Process Flow of Proposed System**

Initially, the data are distributed across a number of separated machines (or virtual machines). From these different partitions, Particles Initialization step collects the initial data to form the initial particles.

During Phase 1, HPSO clustering is constructed on MapReduce by dividing Map step and Reduce step. In Map step, all data in each partition are assigned to the nearest particle according to their Euclidean distances of quasi-attributes. The Euclidean distance measures in HPSO can only calculate from numerical values.

Therefore, all categorical quasi-attributes need to transform into their respective numerical values as in [15]. When the Map step finishes, each particle calculates and updates its fitness, velocity and best position in Reduce step. Here, each particle of swarm can be seen as a Reduce of MapReduce structure. After that, a weakest particle, i.e. the particle with minimum number of data members, is searched and consumed by its nearest strong particle. These Map and Reduce steps iteratively execute until the number of data members in every particle exceed the predefined k-number to attain k-anonymity. The resulted particles can be seen as the intermediate β-clusters.

When Phase 1 finishes, the Phase 2 is stared with a Map step. Here, this Map step does simply by passing all data members of each β-cluster to its respective Reduce step. The Reduce step in Phase 2 runs the normal HPSO clustering job to produce the small data clusters from the large β-cluster. The results of Phase 2 are data clusters with similar quasi-identifiers values that are then generalized to form their anonymized forms.

## V. EXPERIMENTAL SET UP AND RESULTS

We test out approach on the Hadoop cluster of 4 virtual machines with 1GB memory and one virtual CPU. A standard UCI Adult dataset is used to test our approach. It consist of 48842 instances with 14 attributes of both categorical, numerical attributes. From these, we uses 9 attributes (6 categorical and 3 numerical) as quasi-identifiers and 3 attributes (1 categorical and 2 numerical) as sensitive data. 10000 data records of Adult dataset are used for testing our approach.

The experiments are done to analyze the parameters of HPSO and their effects on the utility and privacy of proposed method. To define the information loss, i.e. utility loss, the metric of ILoss [16] is applied on the tested anonymized data.

Our approach is implemented in Java and tested with data of various sizes, ranging from 300 to 10000 data records. The anonymity parameter k is set as 10, and $w^s$=0.5 for weight of proximity, and the number of partition t that varies to makes the sizes of immediate β-clusters in proportion to numbers of data records to 100 for 300, 500, 1000 records, 500 for 3000, 5000, 7000 records, and 1000 for 10000 data records. The numbers of initial particles p are set from 5 to 30 and reduced to its half in proportion to numbers of data records and sizes of immediate β-clusters.

Figure 4 describes the analysis of the proposed system by means of its execution times, iLoss, and varying anonymity parameter k as 5, 10, and 20. Other parameters are given as the above experiment.

By varying anonymity parameter k, the proposed system is analyzed its execution time in seconds and iLoss values. Fig. 4(a) shows that the execution of all testing are almost identical because the sizes of β-clusters are the same. According to Fig. 4(b), we can see that as anonymity parameter k value increases, iLoss value also increases, and that is also for utility loss. This fact indicates that k value must be as less as it can to attain data utility, without regarding the execution time.

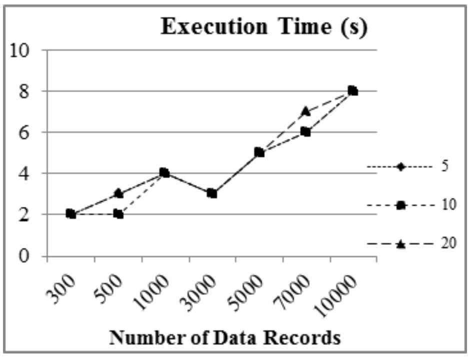

**Fig. 4(A): Analysis of Proposed System's Execution Time(s) and Anonymity parameter *k***

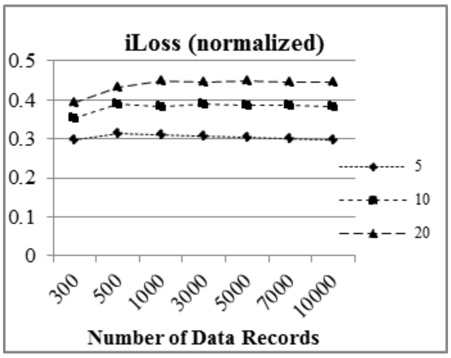

**Fig. 4(B): Analysis of Proposed System's iLoss and Anonymity parameter *k***

## VI. CONCLUSIONS

In this paper, we propose an approach for privacy of big data. While maintaining privacy by k-anonymity, utility loss is also kept into account. Another important fact that needed to consider is execution time of anonymization methods. We try to balance all these factors by constructing HPSO clustering based data anonymization on MapReduce Hadoop infrastructure for big data. The execution time weakness of HPSO clustering is reduced by implementing each particle of HPSO as a Reduce of phase 1 in our approach. The information loss is mostly stable while increasing the data sizes by carefully defining the number of initial particles and anonymity parameter k.

## ACKNOWLEDGMENT

This research is partially supported by University of Computer Studies, Yangon (UCSY), Myanmar, and Fujian University of Technology (FJUT), China.

## *REFERENCES*

[1] Zhang, X., Yang, C., Nepal, S., Liu, C., Dou, W., Chen, J.: A MapReduce Based Approach of Scalable Multidimensional Anonymization for Big Data Privacy Preservation on Cloud. In: IEEE Third International Conference on Cloud and Green Computing, pp. 105--112. IEEE Press (2013)

[2] Zhang, X., Dou, W., Pei, J., Nepal, S., Yang, C., Liu, C., Chen, J.: Proximity-Aware Local-Recoding Anonymization with MapReduce for Scalable Big Data Privacy Preservation in Cloud. IEEE Trans. Computers, vol. 64, no. 8, pp. 2293--2307 (2015)

[3] Upmanyu, M., Namboodiri, A.M., Srinathan, K., Jawahar, C.V.: Efficient Privacy Preserving K-Means Clustering. In: Intelligence and Security Informatics, Lecture Notes in Computer Science, vol. 6122, pp. 154--166. Springer, Berlin Heidelberg (2010)

[4] Rajalakshmi, V., Mala, G.S.A.: Anonymization based on Nested Clustering for Privacy Preservation in Data Mining. J. Computer Science and Engineering (IJCSE). vol. 4, no.3, pp. 216--224, (2013)

[5] Lin, J.L., Wei, M.C.: Genetic algorithm-based clustering approach for k-anonymization. J. Expert Systems with Applications, vol. 36, pp. 9784--9792 (2009)

[6] Bhaladhare, P.R., Jinwala, D.C.: A Clustering Approach for the *l*-Diversity Model in Privacy Preserving Data Mining Using Fractional Calculus-Bacterial Foraging Optimization Algorithm. J. Advances in Computer Engineering, vol. 2014, (2014)

[7] Yin, S., Kaynak, O.: Big Data for Modern Industry: Challenges and Trends. In. Proceedings of the IEEE, vol. 103, no. 2, pp. 143—146 (2015)

[8] Li, T., Li, N.: On the Tradeoff Between Privacy and Utility in Data Publishing. In: KDD'09, Paris, France (2009)

[9] Manta, A.: Literature Survey on Privacy Preserving Mechanisms for Data Publishing. M.S. thesis, Dept. Intelligence Systems, Delft University of Technology, Delft, Netherland, (2013)

[10] Sweeney, L.: k-Anonymity: A Model for Protecting Privacy. In: International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, vol. 10, no. 5, pp. 557--570 (2002)

[11] Machanavajjhala, A., Gehrke, J., Kifer, D., Venkitasubramaniam, M.: ℓ-Diversity: Privacy Beyond k-Anonymity. In: Proc. 22nd International Conference on Data Engineering Workshops (ICDEW'06), (2006)

[12] Ghazi, M.R., D.: Hadoop, MapReduce and HDFS: A Developers Perspective. J. Procedia Computer Science, vol. 48, pp. 45--50 (2015)

[13] Alam, S., Dobbie, G., Riddle, P., Naeem, M.A.: Particle Swarm Optimization Based Hierarchical Agglomerative Clustering. In: IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, pp. 64--68. (2010)

[14] Alam, S., Dobbie, G., Koh, Y.S., Riddle, P., Rehman, S.U.: Research on particle swarm optimization based clustering: A systematic review of literature and techniques. J. Swarm and Evolutionary Computation, vol. 17, pp. 1--13 (2014)

[15] Nouaouria, N., Boukadoum, M.: A Particle Swarm Optimization Approach to Mixed Attribute Data-Set Classification. In: IEEE (2011)

[16] Xiao, X., Tao, Y.: Personalized Privacy Preservation. In: ACM SIGMOD Int'l Conf. Management of Data (SIGMOD'06), pp. 229--240 (2006)