Kronecker Product and Bat Algorithm Based Coefficient Generation for Privacy Protection on Cloud

Nandkishor Karlekar
Vel-Tech Dr. RR & Dr.SR Technical University, Chennai
India
karlekarnandkishor@gmail.com

Gomathi. N
Vel-Tech Dr. RR & Dr.SR Technical University, Chennai
India
gomathin@veltechuniv.edu.in

ABSTRACT: *Due to widespread growth of cloud technology, virtual server accomplished in cloud platform may collect useful data from a client and then jointly disclose the client's sensitive data without permission. Hence, from the perspective of cloud clients, it is very important to take confident technical actions to defend their privacy at client side. Accordingly, different privacy protection techniques have been presented in the literature for safeguarding the original data. This paper presents a technique for privacy preservation of cloud data using kronecker product and Bat algorithm-based coefficient generation. Overall, the proposed privacy preservation method is performed using two important steps. In the first step, PU co-efficient is optimally found out using PUBAT algorithm with new objective function. In the second step, input data and PU co-efficient is then utilized for finding the privacy protected data for further data publishing in cloud environment. For the performance analysis, the experimentation is performed with three datasets namely, cleveland, switzerland and Hungarian and evaluation is performed using accuracy and DBDR. From the outcome, the proposed algorithm obtained the accuracy of 94.28% but the existing algorithm obtained only the 83.64% to prove the utility. On the other hand, the proposed algorithm obtained DBDR of 35.28% but the existing algorithm obtained only 12.89% to prove the privacy measure.*

Keywords: Cloud Computing, Privacy Preservation, Data Publishing, Utility, Accuracy

Received: 23 April 2017, Revised 29 May 2017, Accepted 2 June 2017

1. Introduction

Cloud computing is considered as an inventive merging of technologies and ideas, which establish a pay-as-you-go business model by providing IT services utilizing economies of scale. [13, 16, 17]. This relatively new business model provides benefits to the cloud computing business chains participants since they used to save large IT capital investment using cloud services which includes notable storage and computation capabilities and therefore they more focus well on their heart business [14]. Furthermore, cloud computing presents some wonderful features for science applications in academic world [15]. Also, as cloud

is a multi-tenant environment, the cloud users can easily share data and they can work together as a group. So, many organizations have developed IT systems for their services in cloud computing environments.

Cloud hopeful platform delivers information infrastructure computing and it is considered as a new and resources as IT services [8]. As an example, cloud clients may use these services to perform their tasks in a pay-as-you-go way, at the same time, saving large capital investment in their own IT infrastructure [9]. On the other hand, frequent concern is arising among the customers about their privacy, i.e., whether their private information is safeguarded while using IT services on cloud as there is no control inside the cloud [10]. If there is no clear privacy protection, the customers will ultimately drop the belief and they will not use cloud computing [11]. Thus, privacy protection is a major problem in cloud computing. Privacy protection means the individual's personal information called as sensitive data must be preserved during data publishing [7]. There are three main types of privacy protection methods [12]. First one is perturbative methods, which establish certain type of change into every element of the original data. The second method is generalization methods which swap the original values with less precise ones, and the third one is synthetic data generators which generate synthetic data that looks like the original data [1].

In contrast, the techniques like data distortion, data sanitation, blocking, cryptographic [20], and anonymization [19, 18,] are some of the privacy preserving techniques used to make sure that the privacy of individuals are safeguarded when mining the sensitive data. Many researchers provided incredible research on privacy protection applicable to cloud. Public auditability authentication on cloud needs a high standard to preserve privacy by data provable secure storage [21]. Likewise, data verification in cloud needs to be highlighted in terms of data provability [22]. The above literatures show that there are several privacy protection situations on cloud which can be performed by several privacy protection techniques. Together with those methods, encryption-based methods are also having a main role to protect sensitive data [23, 24]. In [23], Yuan and Yu encrypt the biometric database before outsourcing it to the cloud, and kNN search is performed in the encrypted database. Li *et al.* [24] influence Hierarchical Predicate Encryption to set up a structure for authorized private keyword search on cloud data. Several methods for privacy protection in cloud data are provided by the researchers but still there is a challenge in proper balancing of privacy and utility.

In this paper, PU-bat algorithm is proposed for generating the privacy protected data for data publishing At first, input data which has mixed attributes is directly given to the privacy enabling process where, data matrix and PU (Privacy and Utility) co-efficient are multiplied through Kronecker product. Here, PU co-efficient are new co-efficient defined in the proposed work to multiply with original data matrix. The derivation of PU co-efficient handling both privacy and utility is formulated as a searching problem. So, a recent optimization algorithm called, BAT algorithm [25] is utilized to find out the optimal PU co-efficient by equally considering the privacy and utility. The paper is organized as follows: Section 2 presents the review of literature send section 3 presents the motivation behind the approach. The proposed privacy protection technique is explained in section 4and the results are discussed n section 5. Finally, the conclusion is given section 6.

2. Literature Review

Table 1 discusses the review of recent works presented for privacy preservation in cloud data. Here, the major contributions of the algorithms and their own merits and demerits are tabulated. From the table 1, optimization approaches are presented in [4, 5] for generating privacy protected data. The technique given in [6] generates the noisy data for privacy preservation and some approaches [7] utilized the anonymization techniques for the privacy protection.

3. Motivation behind the Approach

3.1 Privacy-Preserving Outsourcing Model

The proposed work considers the privacy preserving outsourcing model given in figure 1 for data publishing. Here, data providers protect their data through the privacy protection algorithm and the protected data is forwarded to the data storage broker of cloud computing platform. Data storage broker is further communicated with the cloud data management system to store the data within virtual server or physical server in secure way. For getting the information from the database, request handler obtain the users' request and query mapper further match the query within the database to retrieve the relevant information which is then sent to the client who request the service. The major challenge here is the development of privacy protection algorithm for protecting the privacy information without compromising the utility. So, the challenge of transforming the original data, X into the privacy protected data Y requires effective algorithm which equally considers the privacy and utility to find useful for both data providers and clients.

Authors	Contribution	Advantages	Disadvantages
Javier Herranz *et al.* [1]	distance-based record linkage	Handle the privacy and utility effectively	disclosure risk is still presented
Yogachandran Rahulamathavan *et al.* [2]	privacy-preserving (PP) data classification technique using SVM	Multiple encryption have been used	Less adaptive to generation of support vectors
Xuyun Zhang *et al.* [3]	quasi-identifier index based approach	It provides more for economic benefits and operational Convenience	Challenge on processing of huge-volume
Wei Wang *et al.* [4]	optimal sanitization protocol for high dimensional data	Protect again the collusion of providers and users	missing the trade-off between privacy protection and information utility
Fabian Laforet *et al.* [5]	optimization-based privacy preservation	considering several constraints at a time	It considers randomly generated the associations between column values of a bucket
Gaofeng Zhang *et al.* [6]	Time-Series Pattern Based Effective Noise Generation	Without require any assistance from the service providers	Issue in threat noise obfuscation
M. Prakash et al. [7]	personalized anonymization approach	considered only quasi identifier	It does not suitable to multiple sensitive attributes

Table 1. Literature review

3.2 Challenges

The widespread growth of cloud technology makes the involvement of various malicious service providers due to the openness and virtualization nature of cloud. These virtual characteristics may collect useful data from a client and then jointly disclose the client's sensitive data without permission. Hence, from the perspective of cloud clients, it is very important to take confident technical actions to defend their privacy at client side.

Most of the privacy protection methods on data usually concentrate on a single sensitive attribute, or support only low-dimensional data due to the curse of dimensionality. But, in reality, all the attributes have some sensitive information which should be protected and at the same time, curse of dimensionality should be considered.

Existing privacy preservation methods based on shuffling, partitioning, addition of noise information works better for the numerical attributes but the cloud data contains numerical attribute and categorical attributes or mixed in nature. So, a key challenge with privacy protection is dealt with various types of attributes, such as numerical attributes (with real values), categorical attributes (with unranked nominal values), and mixed attributes.

The partially protected data is easier to an adversary for deriving data information belonging to some clients' through the help of the quasi-identifier.

The major challenge is to do privacy preservation over the data without violating utility. When the data is published to third party, it should have some useful information but the sensitive information should be hidden. So, the technique should handle privacy and utility tradeoff effectively.

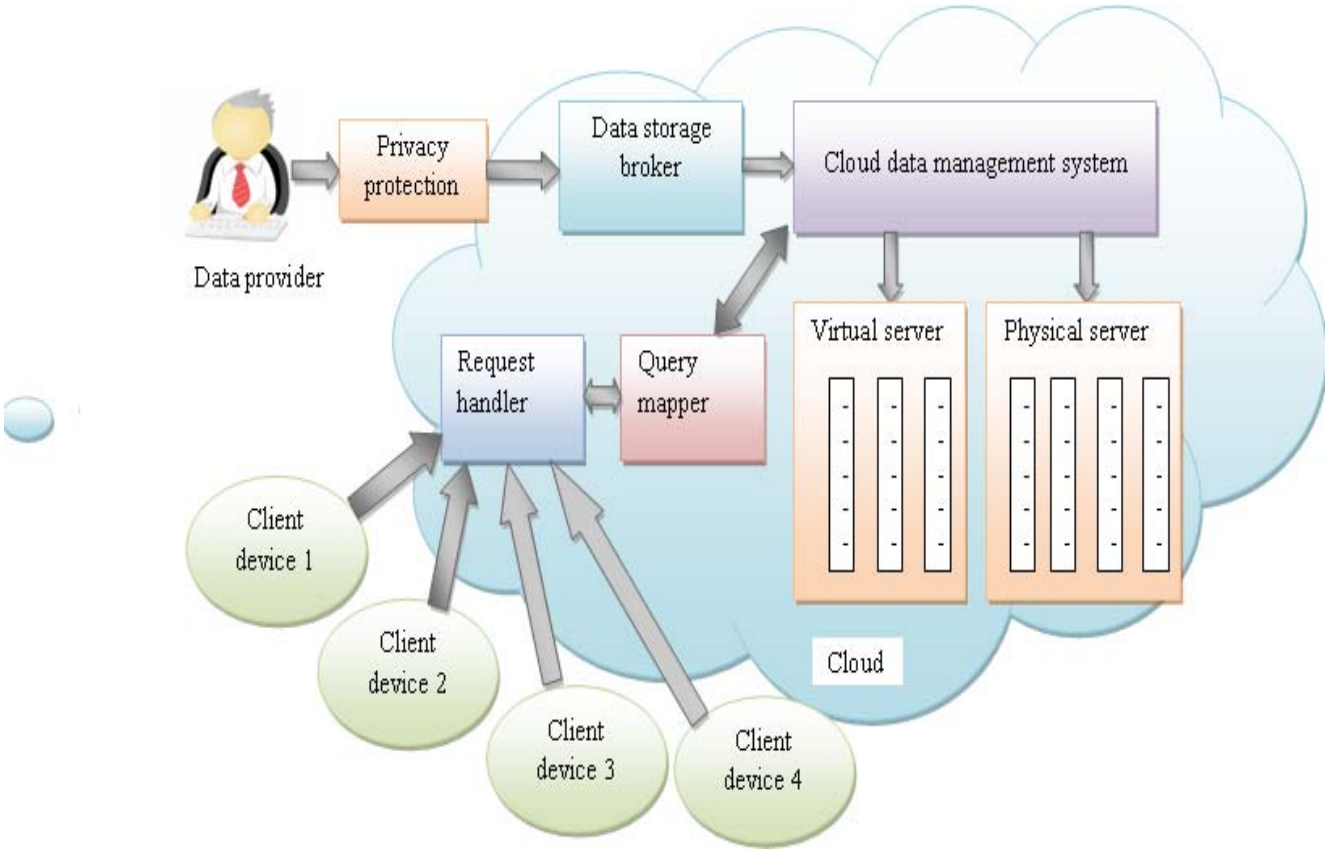

Figure 1. Privacy-preserving outsourcing model in cloud environment

4. Proposed Methodology: Kronecker Product and Bat Algorithm Based Coefficient Generation for Privacy Protection on Cloud

This paper presents a technique for privacy preservation of cloud data using kronecker product and Bat algorithm-based coefficient generation. At first, input data which has mixed attributes is directly given to the privacy enabling process where, data matrix and PU (Privacy and Utility) co-efficient are multiplied through Kronecker product. Here, PU co-efficient are new co-efficient defined in the proposed work to multiply with original data matrix. The derivation of PU co-efficient handling both privacy and utility is formulated as a searching problem. So, a recent optimization algorithm called, BAT algorithm [25] is utilized to find out the optimal PU co-efficient by equally considering the privacy and utility. Overall, the proposed privacy preservation method is performed using two important steps. In the first step, PU co-efficient is optimally found out using PUBAT algorithm with new objective function. In the second step, input data and PU co-efficient is then utilized for finding the privacy protected data for further data publishing in cloud environment.

4.1 Proposed Kronecker Product-based Data Protection

The proposed kronecker product-based data protection scheme is discussed in this section. The idea of transforming the original data X into privacy protected data Y is formulated as mathematical formula as follows:

$$Y = (D \times F) + (H \times E) \quad (1)$$

Where, $\times$ is matrix multiplication, $+$ is element wise matrix addition, Y is the privacy protected data of size $n \times l$ which is exactly the same size of the input data, X. D is correlation matrix of size $n \times (x * n)$, F is PU-enabled data matrix of $(x * n) \times l$, H is PU-enabled data matrix of $n \times (x * l)$ and E is correlation matrix of $(x * l) \times l$.

The PU-enabled data matrix F is computed by performing the kronecker product in between the original data X and the PU-

Coefficient matrix, *C*. PU-coefficient matrix is a matrix which should be generated in more optimal way to handle the privacy of the data publisher and the utility of the client. The data transformation is completely based on the PU- coefficient matrix because it is the core component of the data transformation.

$$F = C \otimes X \tag{2}$$

Where, $\otimes$ is kronecker product, X is input data of $n \times l$ which means that the input data contains n data objects and l attributes, C is PU co-efficient matrix of size $x \times 1$ which will be optimally found out using BAT algorithm, F is PU-enabled matrix of $(x * n) \times l$.

The PU-enabled data matrix of H is computed by performing the kronecker product in between the transpose of PU-coefficient matrix and the original data matrix.

$$H = C^{T'} \otimes X \tag{3}$$

Where, $\otimes$ is kronecker product, C^T is transpose of the PU-coefficient matrix of size $1 \times x$, H is the PU-enabled matrix of $n \times (x * l)$. Then, correlation matrix of D is computed by finding the correlation among the every data object from the original data input X and the PU-enabled matrix F. The objects wise computation of correlation gives the matrix of $n \times (x * n)$ because the input data X contains n objects and the PU-enabled data matrix contains $x * n$ data objects.

$$D = r_{XF}(X, F) \tag{4}$$

Where, D is correlation matrix of size $n \times (x * n)$. Then, correlation matrix of E is computed by finding the correlation among the every data attributes from the transpose of PU-enabled matrix H^T and the original data input X. The attribute wise computation of correlation gives the matrix of $(x * l) \times l$ because the transpose of PU-enabled data matrix H^T contains $x * l$ data attributes X and input data contains l attributes.

$$E = r_{HX}(H^T, X) \tag{5}$$

Where, H^T is transpose of PU-enabled data matrix of size $(x * l) \times n$ and E is a correlation matrix of size $(x * l) \times l$.

4.2 Multi-Objective Optimization Formulation for Privacy Fitness Score

The objective evaluation of the PU-coefficient matrix C is performed using the fitness function defined in the section. As we discussed earlier, PU coefficient matrix is a key component in the privacy protection so the selection of right elements in PU coefficient should satisfy the privacy constraints and utility constraints. The privacy constraint means that the original data should be modified to avoid the inference of guessing the original data values. Utility factor means that the usage of the data after publishing should be more. So, the balance of these two objectives are handled using the proposed objective function.

$$Fitness = \alpha \cdot f_1 + \beta \cdot f_2 + \gamma \cdot f_3 \tag{6}$$

Where, α, β and γ are weighted constants, f_1 is the objective function developed based on the accuracy, f_2 is the objective function developed based on the error, f_3 is the objective function developed based on the difference in original and protected database. The objective function f_1 is computed to consider the utility of information which is found by training the naïve bayes classifier [27] with the original data. The classification accuracy is computed to evaluate the utility. The definition is given as follows.

$$f_1 = \frac{T}{n} \tag{7}$$

Where, T is the number of data objects correctly classified, and is the total number of data objects. The second objective is to

find the error value to ensure the utility of the data. The error values are computed by finding the absolute difference between the ground truth label and the output label which is obtained by the classifier. In order to maximize the overall objective, the cumulative difference is subtracted from unity to maximize this function f_2.

$$f_2 = 1 - \sum_{i=1}^{n} \frac{|O_i - G_i|}{K * n} \tag{8}$$

Where, K is the number of classes, O_i is output of the classifier, G_i is the ground truth and n is the total number of sample cases. The third objective is computed by finding the database difference in between the original database and the protected database. This function is used to ensure the privacy of the publisher. If the difference is high, the privacy will be more. The function f_3 is defined as follows,

$$f_3 = \frac{\sum_{i=1}^{n}\sum_{j=1}^{l}|x_{ij} - y_{ij}|}{n \times l * M_D} \tag{9}$$

$$M_D = \underset{i,j}{\overset{n,l}{Max}}|x_{ij} - y_{ij}| \tag{10}$$

Where, x_{ij} is the tuple value in the original data, y_{ij} is the tuple value in perturbed data, M_D is the maximum difference, and n is the total number of sample cases, l is the number of attributes.

4.3 Bat Algorithm Based PU-coefficient Generation

The finding of the suitable PU-coefficient to satisfy both privacy and utility is a NP hard problem which should be solved using the search algorithm. Here, we utilized a recent search algorithm called, BAT algorithm [25] for finding the suitable PU coefficient. The reason of selecting Bat algorithm is that it has the capability very speedy convergence at a very initial stage by switching from exploration to exploitation. The process of finding the PU coefficient matrix using bat algorithm is given below:

Solution Encoding: The first step in bat algorithm is to encode the solution in efficient way to determine the best solution without much computation overhead. Here, every bat is a vector which contains the x number of element as we now that the size of the PU coefficient matrix is $1 \times x$.

Initialization: The problem space is represented a S in the bat algorithm and it contains the b vectors (bat). Every vector elements is known as the position of the bats. The problem space of $b \times x$ is initialized with random values in the first iteration.

$$S = \{s_i ; 1 \le i \le b\} \tag{11}$$

$$S_i = \{s_{ij} ; 1 \le j \le x\} \tag{12}$$

Where, x is the dimension of the solution. The variable such as, loudness L, pulse rate U, minimum frequency R_{min}, maximum frequency R_{max} and velocity v are also initialized.

Evaluation: The next step is to evaluate the position vector of bat (PU coefficient vector) to find the fitness of the vector. The fitness function is already explained in the above section. The solution vector which is having the maximum fitness is stored as a separate variable called, S_{max}.

Movement of Virtual Bats: Every bat is then updated their position using the frequency and velocity with the following

equation. The equation utilized in BAT algorithm is as follows,

$$R_i = R_{\min} + \left(R_{\max} - R_{\min}\right) * \delta \tag{13}$$

$$v_i^t = v_i^{t-1} + R_i\left(s_i^{t-1} - s_{\max}\right) \tag{14}$$

$$s_i^t = s_i^{t-1} + v_i^t \tag{15}$$

Where, δ is random value which is used to update the frequency of the bat using minimum R_{min} and maximum frequency R_{max}. It ranges between -1 to 1.The frequency R_i is then utilized to update the velocity of the bats (v_i^t) using best position of the bats S_{max}. Then, position of the bats is computed using the velocity and position of the last iteration.

Loudness and Pulse Rate-based Movement: In this step, random value δ is generated and if the random value is greater than the pulse rate, *U*, new local solution is generated based on the best solutions. Also, if the random value is lesser than the loudness, *L*, random solution is generated and the loudness and pulse rate are updated only if this random solution is better than the best solution.

Termination: The process above repeats until all the virtual bats are updated their position. Thus, one generation is finished. The iteration goes until terminal requirement of T_{max} iteration is met. Then, the best is output as the optimal solution to the problem.

5. Results and Discussion

This section presents the experimentation of the proposed PU-bat algorithm and the comparative analysis with existing work with three different datasets.

5.1 Experimental Set up

The proposed algorithm is implemented using Java 1.7 with net beans IDE 7.3. The experimentation is conducted on Windows 8.1 machines with Intel Core i5 processors and 4 GB of main memory. The clouding computing platform is simulated using cloudsim tool and the proposed data protection algorithm is implemented using JAVA and it is incorporated within cloudsim tool. The performance of the proposed algorithm will be compared with other privacy model described in [26].

Dataset Description: The experimentation is performed with three datasets namely, cleveland, switzerland and Hungarian which are obtained from UCI machine learning repository [29]. These data were surveyed by Robert Detrano from V.A. Medical centre. The processed database consists of 76 attributes, but the popular research works make use of a subset of 14 of them. The "class" field indicates the presence of heart disease in the patient. It is integer valued from 0 (no presence) to 4.

Evaluation Metrics: The proposed algorithm is evaluated suing two metrics namely, accuracy and Database difference ratio (DBDR). The utility of the algorithm is evaluated using accuracy.

$$f_1 = \frac{T}{n} \tag{16}$$

Where, *T* is the number of data objects correctly classified, and *n* is the total number of data objects. The privacy of the algorithis evaluated using database difference ratio [28] which is again normalized to range of value within 0 to 1.

$$f_3 = \frac{\sum_{i=1}^{n}\sum_{j=1}^{l}\left|x_{ij} - y_{ij}\right|}{n \times l * M_D} \tag{17}$$

1 **Algorithm: PU-BAT**

2 **Input:** X →Original database

3 α , β , γ → Weighted constants

4 **Output:**

5 $s_{\max}$ → best solution (Optimal PU-coefficients)

6 **Begin**

7 **Initialize** variables such as, L , U , $T_{\max}$, $R_{\min}$, $R_{\max}$, δ

8 **Initialize** $t = 1$, bat population S and velocity v

9 **While** $t < T_{\max}$

10 **Find** fitness for s_i

11 **Update** velocity v_i^t and frequency R_i

12 **Update** bats positions by s_i^t

13 **Store** best solution $s_{\max}$

14 **If**($\delta > U$)

15 **Generate** local solution around best solution

16 **End if**

17 **If**($\delta < L$)

18 **Generate** random solution, s_r

19 **Find** fitness of s_r

20 **If**(fitness(s_r)<fitness($s_{\max}$)

21 **Update** $s_{\max}$, U and L

22 **End if**

23 **End if**

24 $t = t + 1$

25 **End while**

26 **Return** $s_{\max}$

27 **End**

Figure 2. Pseudo code of PUBAT algorithm

$$M_D = \underset{i,j}{\overset{n,l}{Max}} \left| x_{ij} - y_{ij} \right| \quad (18)$$

Where, x_{ij} is the tuple value in the original data, y_{ij} is the tuple value in perturbed data, M_D is the maximum difference, and n is the total number of sample cases, l is the number of attributes.

Parameters to be fixed: The important parameters to be fixed for the proposed algorithm are three weighted constants such as, alpha, beta and gamma which are utilized in the fitness function. These values are analysed with various iterations and the better values suitable for the proposed algorithm is obtained.

5.2 Performance Evaluation of the Proposed Algorithm

The performance of the proposed algorithm is analyzed with the various values of alpha, beta and gamma for various number of iterations. Figure 3 shows the performance graph for the Cleveland data. From figure 3.a, we understand the maximum accuracy is obtained for the maximum number of iterations. Also, when α, β and γ are fixed to 0.5, the accuracy is 94.28% if the iterations are equal to 50. Figure 3.b shows the performance graph of the proposed PU-bat algorithm in terms of DBDR. For the better privacy, the DBDR should be high which means that the protected database have much difference than the original database. Here, the maximum privacy is obtained when α, β and γ are fixed to 0.5. The DBDR obtained for these thresholds is 35.28 which is higher than the other thresholds. So, from this graph, we conclude that the better performance in terms of privacy and utility can be obtained when the weighted constants α, β and γ are fixed to 0.5.

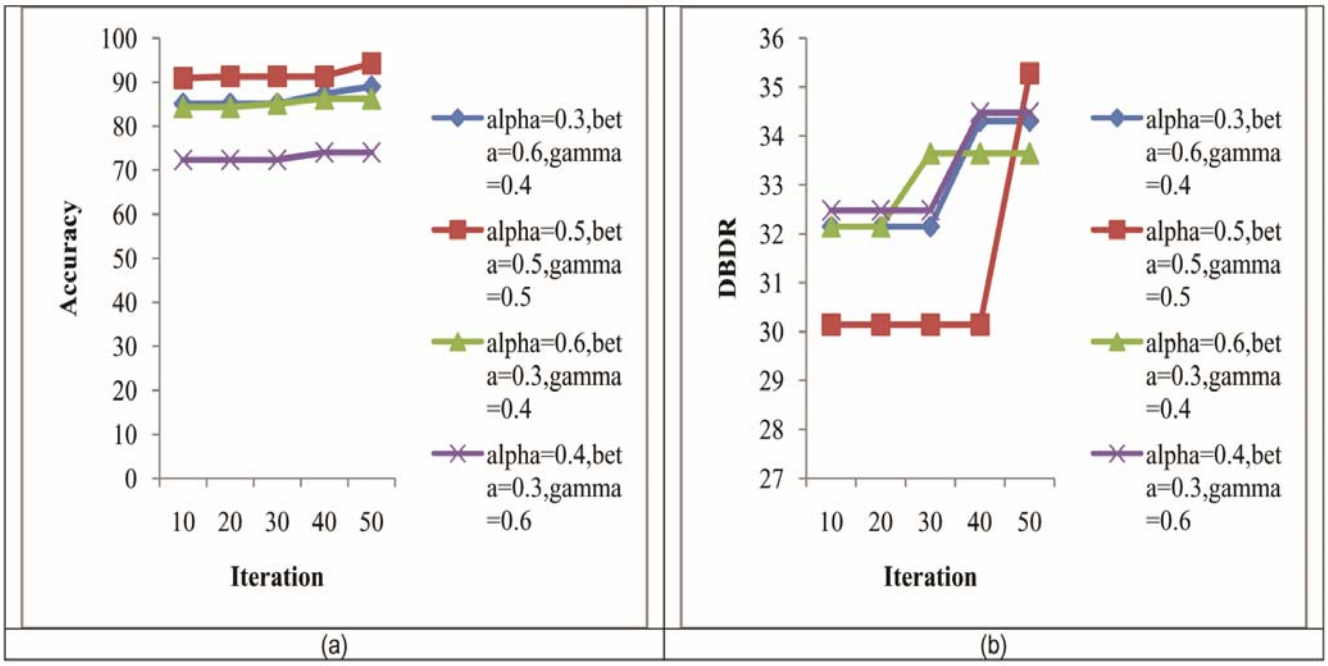

Figure 3. Performance evaluation on cleveland data, a) accuracy, b) DBDR

Figure 4 shows the performance of the proposed PU-bat algorithm using accuracy and DBDR on Switzerland data. From figure 4.a, we understand that the accuracy is improved when the number of iterations is fixed to higher value. In Switzerland data, the higher accuracy of 95.22% is obtained when α, β and γ are fixed to 0.3, 0.6 and 0.4. Also, the accuracy of 93.45% is obtained when α, β and γ and are fixed to 0.5. From the figure 3.b, we understand that the DBDR of the proposed algorithm is 39.43% when α, β and γ and are fixed to 0.4, 0.3 and 0.6. The DBDR is increased for the all the parametric values when iteration is increased.

Figure 5 shows the performance evaluation on hungarian data using accuracy and DBDR. From the figure 5.a, the accuracy values are increased for the different parametric values when the number of iterations is increasing. Here, the higher accuracy of

94.36% is obtained when α, β and γ are fixed to 0.3, 0.6 and 0.4. The accuracy value of 91.39% is obtained when α, β and γ and are fixed to 0.5. Similarly, the performance of the proposed algorithm using DBDR is given in figure 5.b. Here, the higher value of DBDR obtained by the proposed algorithm is 33.63% when α, β and γ and are fixed to 0.6, 0.3 and 0.4. Overall, the average performance is computed for the different parametric values and the better parametric values are selected for the performance comparison with the existing works. From the average performance, the better parametric values for α, β and γ and are 0.3, 0.6 and 0.4.

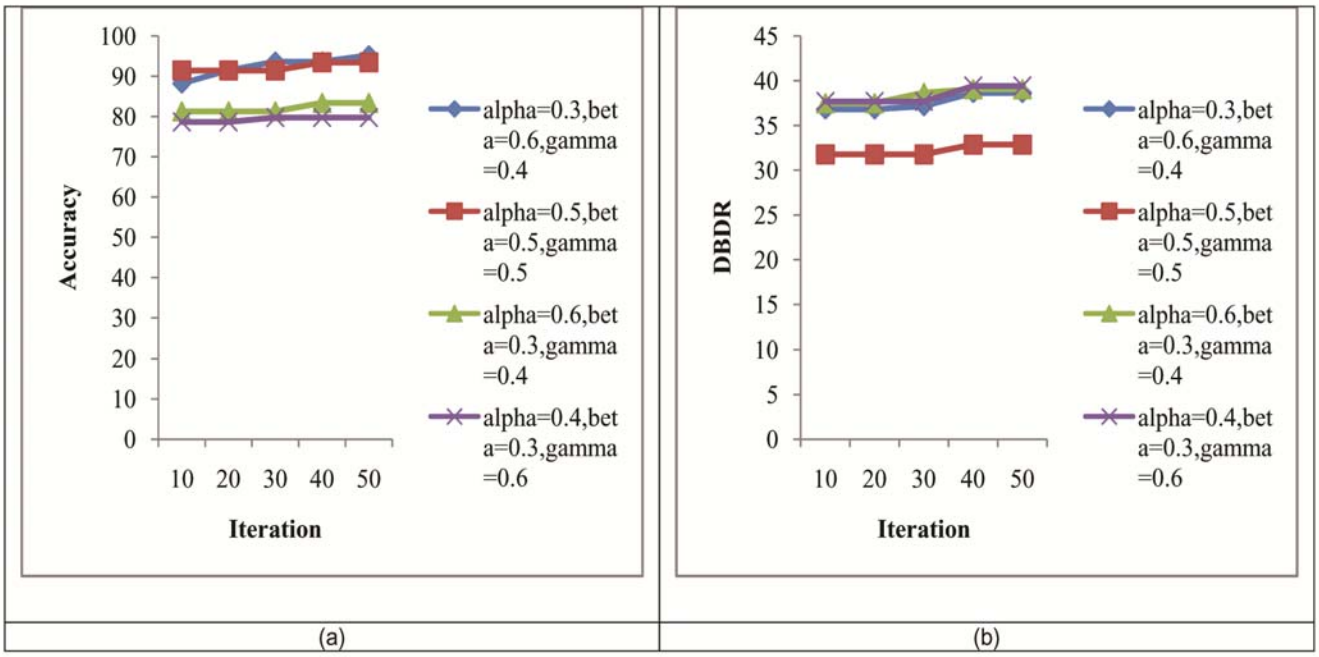

Figure 4. Performance evaluation on switzerland data, a) accuracy, b) DBDR

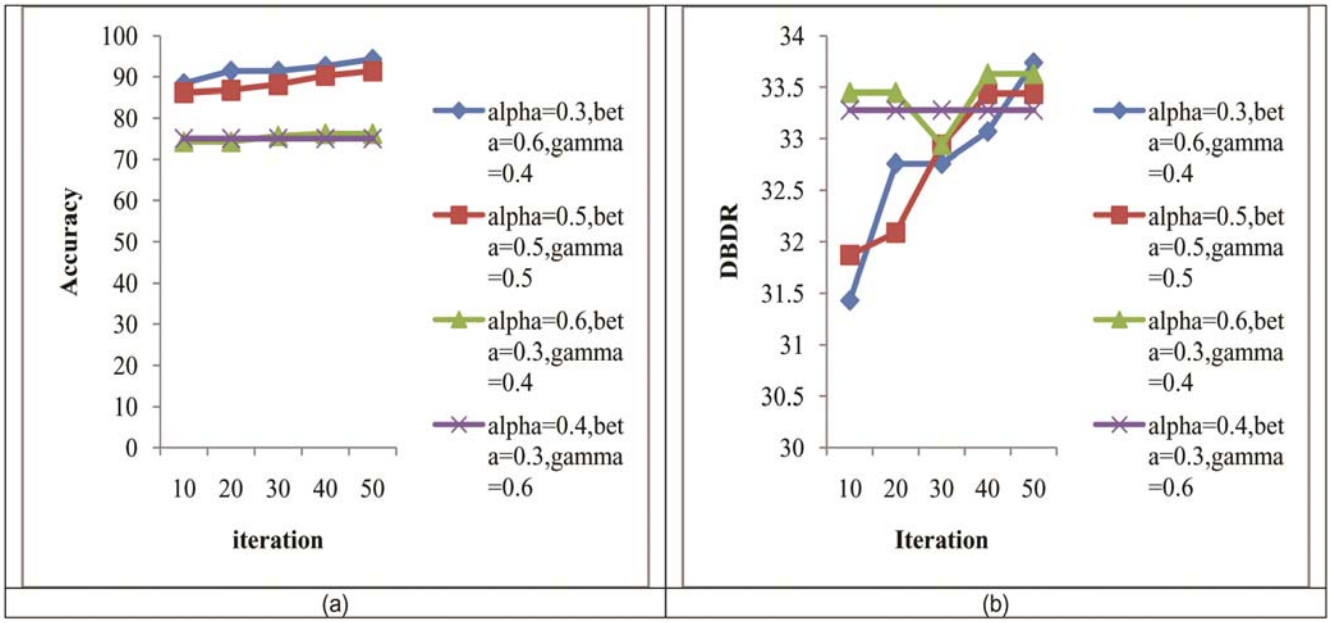

Figure 5. Performance evaluation on hungarian data, a) accuracy, b) DBDR

5.3 Comparative Analysis of the Proposed Algorithm

This section presents the comparative analysis of the proposed algorithm with the existing work given in [26]. The performance is compared using accuracy and DBDR on three different datasets taken for experimentation. Figure 6.a shows the comparative analysis on cleveland data using accuracy. Here, the performance of both the algorithm is compared for the different percentage of data given for experimentation. For all the percentage of data given for experimentation, the proposed algorithm outperformed the existing algorithm in both the metrics, accuracy and DBDR. When the percentage of data is 100, the proposed algorithm obtained the accuracy of 94.28% but the existing algorithm obtained only the 83.64%. Also, figure 6.b shows the performance comparison of both the algorithms using DBDR. Here, the better performance of 35.28% is obtained when the percentage of data is fixed to 100.

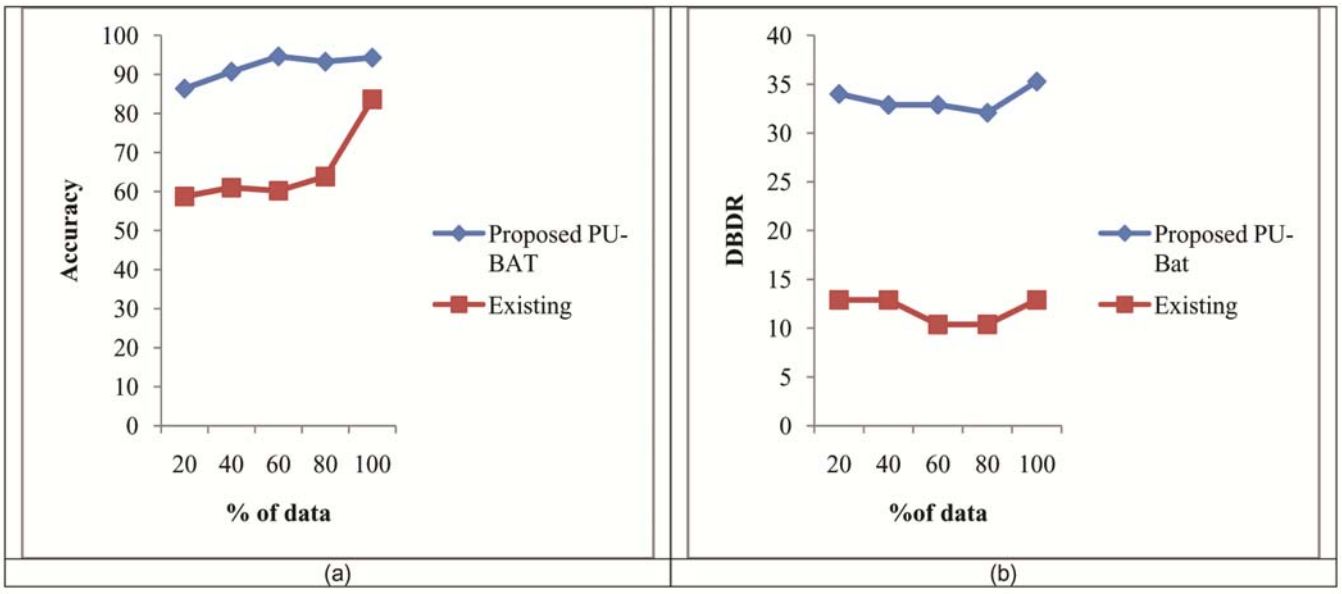

Figure 6. Comparative analysis on cleveland data, a) accuracy, b) DBDR

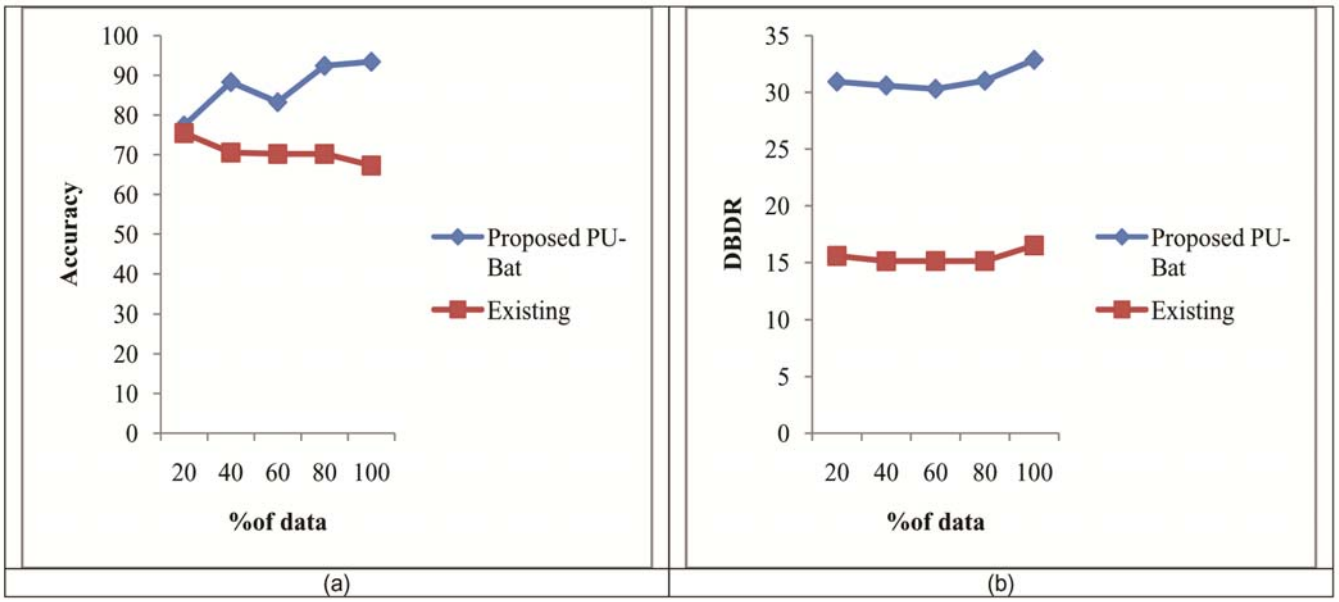

Figure 7. Comparative analysis on switzerland data, a) accuracy, b) DBDR

Figure 7 shows the performance comparison of the proposed algorithm with the existing algorithm on Switzerland data. Here, the accuracy values are constant when the percentage of data is fixed to 20 but other percentage of training data, the proposed algorithm outperformed the existing algorithm. The higher accuracy of 93.45% is obtained by the proposed algorithm when the percentage of data is 100%. Figure 7.b shows the performance comparison of the proposed algorithm on Hungarian data using DBDR. Here, the better performance is obtained by the proposed algorithm for all the different percentage of data compared with the existing algorithm. The higher DBDR value of 32.87% is obtained by the proposed algorithm while compared with the existing algorithm which has achieved only the 16.15%.

Figure 8 shows the comparative analysis on hungarian data for both the algorithms. From figure 8.a, the accuracy of the proposed algorithm is 93.46%, 91.87%, 90.05%, 90.18% and 91.39% when percentage of data is 20% to 100%. Similarly, the existing algorithm obtained the accuracy of 74.92%, 70.07%, 69.74%, 69.74% and 66.72% when the percentage of data is 20% to 100%. Figure 3.b shows the performance of the algorithms on DBDR. Here, the proposed algorithm obtained the higher DBDR value of 35.41% but the existing algorithm obtained the value of 15.33%. Overall, the proposed algorithm outperformed the existing algorithm both in accuracy and DBDR on three datasets.

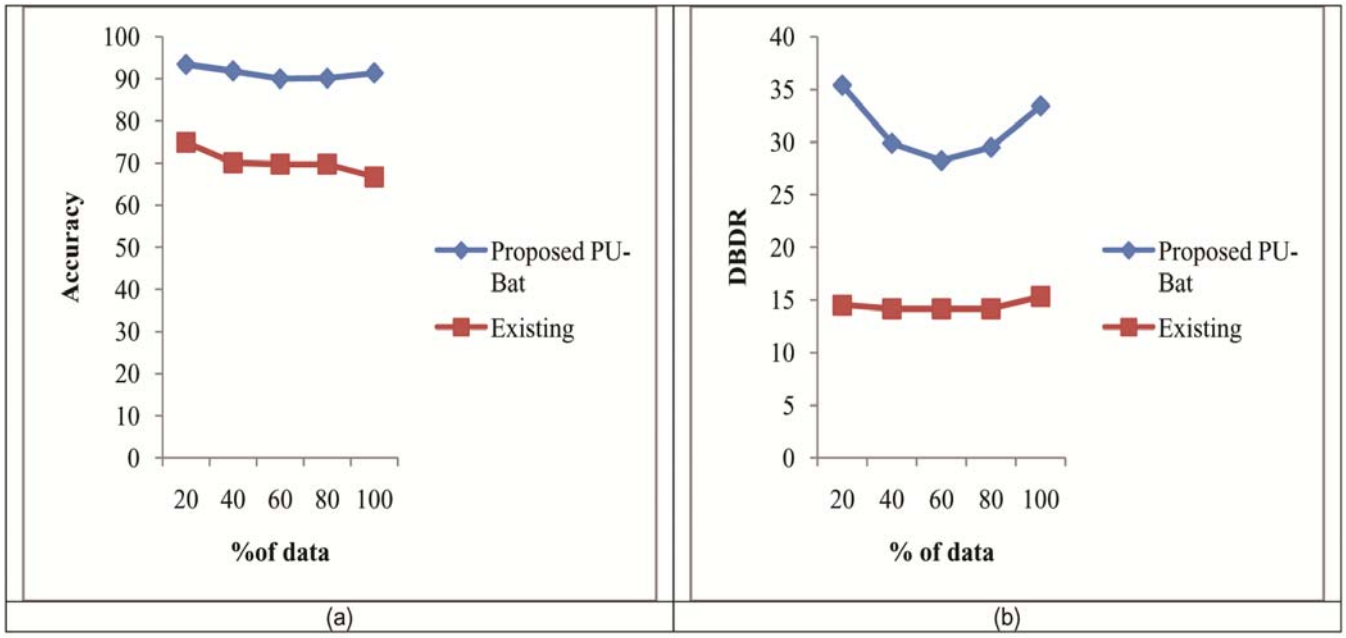

Figure 8. Comparative analysis on hungarian data, a) accuracy, b) DBDR

6. Conclusion

We have presented a PU-bat algorithm for generating the privacy protected data for data publishing. Here, the input data was directly given to Bat algorithm for finding the optimal PU-coefficients using the proposed multi-objective function. The proposed multi-objective function considers three objective constraints to balance the privacy and utility. Then, the optimal PU-coefficients are utilized to generate the privacy protected data using kronecker product. To ensure the applicability of the proposed algorithm, the experimentation is performed with three datasets and quantitative measurement is performed using accuracy and DBDR. From the outcome, the proposed algorithm obtained the accuracy of 94.28% but the existing algorithm obtained only the 83.64% to prove the utility. On the other hand, the proposed algorithm obtained DBDR of 35.28% but the existing algorithm obtained only 12.89% to prove the privacy measure. In future, Bat algorithm can be replaced with the hybrid algorithms for finding the optimal coefficient in quicker time.

References

[1] Herranz, Javier., Nin, Jordi., Rodríguez, Pablo., Tassa, Tamir. (2015). Revisiting distance-based record linkage for privacy-

preserving release of statistical datasets, *Data & Knowledge Engineering*.

[2] Rahulamathavan, Yogachandran., Cumanan, Kanapathippillai., Rajarajan, Muttukrishnan. (2014). Privacy-Preserving Multi-Class Support Vector Machine for Outsourcing the Data classification in Cloud, *IEEE Transactions on Dependable and Secure Computing*, 11 (5) 467-478.

[3] Zhang, Xuyun., Liu, Chang., Nepal, Surya., Chen, Jinjun. (2013). An efficient quasi-identifier index based approach for privacy preservation over incremental data sets on cloud, *Journal of Computer and System Science*s, 79, 542–555.

[4] Wang, Wei., Chen, Lei., Zhang, Qian. (2015). Outsourcing high-dimensional healthcare data to cloud with personalized privacy preservation", *Computer Networks*, 88, p. 136–148.

[5] Laforet, Fabian., Buchmann, Erik., Böhm, Klemens. (2015). Individual privacy constraints on time-series data, *Information Systems*, 54, p. 74-91.

[6] Zhang, Gaofeng., Liu, Xiao., Yang, Yun. (2015). Time-Series Pattern Based Effective Noise Generation for Privacy Protection on Cloud, *IEEE Transactions on Computers*, 64 (5) 1456-1469, May 2015.

[7] Prakash, M., Singaravel, G. (2015). An approach for prevention of privacy breach and information leakage in sensitive data mining, *Computers and Electrical Engineering*.

[8] Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., Brandic, I. (2009). Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility, *Future Generation Computer Systems*, 25 (6) 599-616.

[9] Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R.H., Konwinski, A., Lee, G., Patterson, D.A., Rabkin, A., Stoica, I., Zaharia, M. (2010). Above the Clouds: A Berkeley View of Cloud Computing, Comm. ACM, 53 (6) 50-58.

[10] Jansen, W., Timothy, G. (2011). Guidelines on Security and Privacy in Public Cloud Computing. Nat'l Inst. Standard and Technology, Special Publication 800-144, (December).

[11] Ryan, M. D. (2011). Cloud Computing Privacy Concerns on our Doorstep, *Comm. ACM*, 54 (1) 36-38.

[12] Torra, V. (2009). Handbook of data mining, chapter Privacy in Data Mining, *Human Factor and Ergonomics*.

[13] Wang, L., Laszewski, Von, G., Younge, A., He, X., Kunze, M., Tao, J., Fu. C. (2010). Cloud computing: A perspective study, New Generation Comput, 28 (2) 137–146.

[14] Wang, L., Zhan, J., Shi, W., Liang, Y. (2012). In cloud, can scientific communities benefit from the economies of scale?, IEEE Trans. Parallel Distrib. Syst, 23 (2) 296–303.

[15] Yang, X., Wang, L., Laszewski, G. (2009). Recent research advances in e-science, *Cluster Comput*., 12 (4) 353–356.

[16] Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., Zaharia, M. (2010). A view of cloud computing, *Commun*. ACM, 53 (4) 50–58.

[17] Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., Brandic, I. (2009). Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility, *Future Generation Comput. Syst*. 25 (6) 599–616.

[18] Li, Tiancheng., Li, Ninghui. (2012). Slicing: a new approach for privacy preserving data publishing, *IEEE Trans Knowl Data Eng*, 24 (3) 561–574.

[19] Inan, A., Saygin, Y., Savas, E., Hintoglu, A., Levi, A. (2006). Privacy preserving clustering on horizontally partitioned data, *In*: Data Engineering Workshop.

[20] Pinkas, B. (2002). Cryptographic techniques for privacy preserving data mining, ACM SIGKDD Explor. 2002.

[21] Wang, C., Chow, S. S. M., Wang, Q., Ren, K., Lou, W. (2013). Privacy Preserving Public Auditing for Secure Cloud Storage, *IEEE Trans. Computers*, 62 (2) 362-375, (February).

[22] Zhu, Y., Hu, H., Ahn, G.-J., Yu, M. (2012). Cooperative Provable Data Possession for Integrity Verification in Multi-Cloud Storage, *IEEE Trans. Parallel and Distributed Systems*, 23 (12) 2231-2244 (December).

[23] Yuan, J. Yu, . S (2013). Efficient privacy-preserving biometric identification in cloud computing, *In:* Proceedings of the IEEE INFOCOM, 2013.

[24] Li, M., Yu, S., Cao, N., Lou, W. (2011). Authorized private keyword search over encrypted data in cloud computing, *In*:

Proceedings of the IEEE ICDCS.

[25] Yang, X. S. (2010). A New Metaheuristic Bat-Inspired Algorithm, in: Nature Inspired Cooperative Strategies for Optimization (NISCO 2010) (Eds. J. R. Gonzalez et al.), *Studies in Computational Intelligence*, Springer Berlin, 284, Springer, 65-74.

[26] Pan, Yang., Gui Xiaolin, Jian, An., Jing, Yao., Jiancai, Lin., Feng, Tian. (2014). A Retrievable Data Perturbation Method Used in Privacy-Preserving in Cloud Computing, *China Communications*, 11 (8) 73-84.

[27] Rish, I. (2001). An empirical study of the naive bayes classifier, IJCAI 2001 workshop on empirical methods in artificial intelligence, 3 (22) 41-46.

[28] Yeh, Jieh-Shan., Hsu, Po-Chiang. (2010). HHUIF and MSICF: Novel algorithms for privacy preserving utility mining , *Expert Systems with Applications*, 37, p. 4779–4786.

[29] UCI machine learning repository from https://archive.ics.uci.edu/ml/datasets/Heart+Disease