

# An Adaptive Hidden Markov Model with CCA for Privacy Preserving of Correlated Big Data

Sujatha K<sup>1</sup>, Rajesh N<sup>2</sup>

<sup>1,2</sup>Department of Information Technology, University of Technology and Applied Sciences-Shina, Oman

## Article Info

### Article history:

Received May 01, 2022

Revised May 06, 2022

Accepted May 15, 2022

### Keywords:

Differential privacy

Correlated big data

Privacy preserving big data

Hidden Markov Model

Canonical correlation analysis

## ABSTRACT

Due to technological advancements and increase in the use of smart devices, huge amount of data is generated and has open access to various social media servers all around the world. Most of the social media providers seldom care on security and or preservation of private data. One of the greatest challenges that prevail due to the existence of correlated information is privacy preserving data mining. Many research methodologies have been proposed yet maintaining the privacy in social network is a complex process. In this proposed method, a novel methodology for preserving the privacy of Correlated big data using various techniques. The proposed method consists of three main processes they are, correlated big data identification, correlated big data analysis and correlated iteration mechanism. In first process an adaptive Hidden Markov Model (AHMM) used for identifying the Correlated big data present in the datasets. Then in the second process, using the canonical correlation analysis (CCA) the correlation matrix is find out for sensitivity measure. In last process, to answer the large group of queries designed a correlated iteration mechanism. Thus implemented the proposed system and the implementation results are compared with the conventional techniques. Ultimately the proposed method suggests that the performance is better for the privacy preserving of correlated dataset.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



## Corresponding Author:

Sujatha K

Department of Information Technology

University of Technology and Applied Sciences-Shinas

Oman

Email: sujathasjcit@gmail.com

## 1. INTRODUCTION

In recent times, collected data by business and scientific organizations are in substantial quantities and organized in databases. As it results in knowledge discovery analysis of these databases are extremely useful. Because of this, through data mining methods remarkable effort has dedicated to the development of methods that aids knowledge discovery [1]. By means of data mining process discovered the valuable patterns and relationships that lie concealed inside extremely huge databases [2]. In Knowledge Discovery in Databases, continues to be one of the most prevalent pattern-discovery methods for the mining of such rules [3]. Data mining is an important step in the Knowledge Discovery in Databases (KDD) process and it generates a definite enumeration of patterns over the data by employing computational methods subject to tolerable computational efficiency restrictions [4]. Regularly gather and analyze large quantities of comprehensive personal data or the

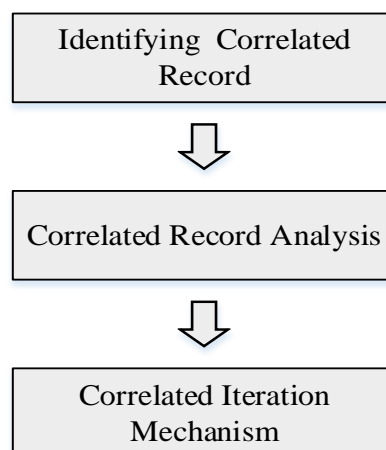
big data through the applications used in the data mining. Shopping behaviors, illegalRecords, therapeutic history, and credit account etc. are few examples for such data [5]. The reason for the emergence of data mining as one of the most significant research fields are due to the expansion of both computer hardware and software technologies [6]. It is obvious that data which is regarded as the most significant benefit of any establishment; should be used for predicting future decision [7]. Normally, the centralized collections of data use data mining algorithms. Certainly, when data is gathered from diverse places the precision of any data mining task may also improved [8]. Importance of Data mining are in Marking/Retailing, Banking/Crediting, Law enforcement, Researchers, Transportation, Medicine, Insurance, improve effectiveness and decrease price [9].

Due to the zooming capacity to amass personal data on clients the perennial paradox of privacy-preserving data mining has assumed alarming proportions in recent years and the burgeoning style of data mining algorithms to control the related data [10]. In favor of a multiplicity of unfair motives [11] this has astonishingly sprouted alarming apprehensions about the impending abuse of personal data. There has been a feast of methods launched of late targeted at executing the data mining functions in a privacy-preserving manner with a view to tackle these dilemmas. The captioned techniques are drawn form an extensive range of allied subjects like data mining, cryptography, and data hiding [12]. The astounding application of data mining devices in both the public and private sectors has cropped up numerous disputes in respect of the potentially susceptible character of a lion's share of the extracted data [13]. A direct conflict with the needs of the clients and their right to confidentiality is the potential advantages from extensive data mining [14]. Privacy preserving data mining algorithms are targeted at designing an inventive data mining technique for the reason of using data unchanged. This directs the clients to enjoy the advantages of data mining with no compromise in their privacy [15]. Noman Mohammed *et al.* [16] have deal with the difficulty of private data publishing, where dissimilar aspects for the similar set of individuals are detained by two parties. Pui K. Fong and Jens H. Weber-Jahnke [17] introduced a privacy preserving approach that can be applied to decision tree learning, without concomitant loss of accuracy. Jaideep Vaidya *et al.* [18] have proposed the framework based on Random Decision Trees in privacy preserving data mining. Xuyun Zhang *et al.* [19] have proposed a scalable two-phase top-down specialization approach to anonymize large-scale data sets using the Map Reduce framework on cloud.

The rest of the paper is ordered as follows: In section 2 gives the proposed methodology for the privacy preserving of Correlated big data. Section 3 given the detail description on implementation results and in the subsequent sections conclusion for this paper and given the list of references.

## 2. PROPOSED METHODOLOGY FOR THE PRIVACY PRESERVING OF CORRELATED BIGDATA

A big challenge is the privacy preserving of correlated big data from the social network so that this work designed to develop a novel procedure for privacy preserving of Correlated big data called as differential privacy. Correlated big data identification, sensitive analysis and correlated iteration mechanism are the three main processes used in the proposed differential privacy system. In the first phase using an adaptive Hidden Markov Model (AHMM) identified the Correlated big data from the big data. In the second phase analyzed the sensitivity of the Correlated big data to find the correlation matrix by using canonical correlation analysis (CCA). In the last phase designed a correlated iteration mechanism to answer the large group of queries, which builds a dataset series to answer all queries by iteratively inform the datasets. Fig 1 shows the process flow of the proposed method.



**Figure 1:** Process flow of the proposed system for differential privacy

**2.1. Identifying Correlated big data**

With big data identifying the actual Correlated big data is really a big undertaking, and to deliver privacy for the Correlated big data is essential. Proposed an Adaptive Hidden Markov Model (AHMM) to discover the Correlated big data, exist inside the data. To identify the coupled behavior from the dataset is the main role of the AHMM, based on this the Correlated big data identified on the big dataset.

**a. Formation of objective**

In case that information has become collected through wide analyses which might be recognized against each other, for case, DNA binding proteins which might be a piece of a necessary protein difficult, or like DNA binding proteins profiled within distinctive treatment conditions or perhaps diverse cell lines.

In the same way, an additional small sample, a gang of criminals performs a further development of behavior to attain their target. The activities are related jointly and have the accurate goal. Each test yields the data series measured all through the genome. Suppose we now have  $N$  series, indicated simply by  $O = \{O_{j,t}\}$  for  $t = 1, \dots, T$  along with  $j = 1, \dots, N$ , where  $O_{j,t}$  denotes the observed datum (emission) on window  $t$ . The observed data  $O$  are associated to binary hidden states  $H = \{h_{j,t}\}$  indicating the position of either true binding site (state 1) or background site (state 0). We also use  $O_t$  and  $h_t$  to signify the definite observed data and hidden states, respectively, on window  $t$  around all series. The target of HMM analysis could be the estimation of  $H$  related with. As before mentioned, while using the two states independent HMM is usually a straightforward strategy, where every series  $j$  comes with an independent HMM. Standard HMM for the single data series has three elements: (i) the probability mass function from the first window  $\pi(h_{j0} = 1)$ ; (ii) the transition kernel  $K_j$  in eq. (1), which is constant for all windows  $\{t : 0 \leq t \leq T\}$ ; and (iii) the emission  $\pi(O_{jt} | h_{jt} = 0)$  and  $\pi(O_{jt} | h_{jt} = 1)$  for all  $t$ . The likelihood of this model might be written such as in eq. (2)

$$K_j = \begin{pmatrix} 1-p_j & p_j \\ 1-q_j & q_j \end{pmatrix} \rightarrow (1)$$

$$L = \prod_{j=1}^N \left\{ \pi(h_{j0}) \pi(O_{j0} | h_{j0}) \cdot \prod_{t=2}^T \pi(h_{jt}) \pi(O_{jt} | h_{jt}) \right\} \rightarrow (2)$$

Where,

$$p_j = \pi(h_{j,t} = 1 | h_{j,t-1} = 0)$$

$$q_j = \pi(h_{j,t} = 1 | h_{j,t-1} = 1)$$

The forward-backward algorithm can be applied from the given three components to deduce the hidden states rapidly. While independent HMM can be fast as well as straightforward, it struggles to capture the correlation regarding associated information series. This correlation can be openly integrated from the model to develop statistical power and eliminate noise. By far the most instinctive technique is HMM, a single HMM together with  $2^N$  hidden state combinations as well as  $2^N \times 2^N$  transition kernel. The likelihood of HMM can be

$$L = \pi(h_0) \pi(O_0 | h_0) \cdot \prod_{t=2}^T \pi(h_t) \pi(O_t | h_t) \rightarrow (3)$$

$$= \pi(h_0) \prod_{j=1}^N \pi(O_{j0} | h_{j0}) \cdot \prod_{t=2}^T \left( \pi(h_t) \prod_{j=1}^N \pi(O_{jt} | h_{jt}) \right)$$

Here the emission is normally assumed as in addition to the hidden states in some other series, i.e.,  $\pi(O_{jt} | h_t) = \pi(O_{jt} | h_{jt})$  for all  $j$  and  $t$ . A lot like independent HMM, the forward-backward algorithm is generally applied to infer the hidden states likewise.

**b. Adaptive Hidden Markov Model**

Even if HMM makes up about the Markovian dynamics between hidden state combinations, not just about all combinations automatically happen inside data as well as, more particularly, the sort of computation for the forward backward algorithm is  $O(T \cdot 4^N)$  in contrast to  $O(N \cdot 4^T)$  in independent HMM. This clearly limits the applicability of HMM to cases along with small or moderate  $N$ . When the objective of the analysis is the inference of hidden states in each series, HMM is computationally incompetent.

This determined us to formulate an AHMM, a compromise between the two methods. Inside the proposed AHMM, it is permitted for dissimilar series to obtain correlated just as HMM. At the same time, two methods are used, which decrease the computational cost to a level comparable to that of independent HMM. First, in place of considering all series simultaneously, iteratively infer variables by cycling through each series individually. For each one series performed the inference i conditioning within the current hidden state vectors in each other series. Next, all through each series assumed the sparsity while incorporating correlations between the current series and all other series. The correlation imposed with the form of an inhomogeneous transition kernel. For being specific, denoted the transition kernel of AHMM for series  $j$  at window  $t$  as in eq. (4).

$$K_j(t) = \begin{pmatrix} 1 - p_{jt} & p_{jt} \\ 1 - q_{jt} & q_{jt} \end{pmatrix} \rightarrow (4)$$

Because the transition possibility differs by means of window  $t$  it has an additional index ( $t$ ). Here, defined  $p_{jt}$  and  $q_{jt}$  to combine the input from other series as follows:

$$p_{jt} = \pi(h_{j,t} = 1 | h_{j,t-1} = 0, \{(h_{k,t-1}, h_{k,t})\}_{k \neq j})$$

$$q_{jt} = \pi(h_{j,t} = 1 | h_{j,t-1} = 1, \{(h_{k,t-1}, h_{k,t})\}_{k \neq j})$$

That is, adjusted the transition probability in series  $j$  because of the hidden states in various other series  $\{l : l \neq j\}$ . Here consider a couple of logistic regression models in each data series:

$$\log\left(\frac{p_{jt}}{1 - p_{jt}}\right) = \beta_{j0}^p + \sum_{k \neq j} (\beta_{jk}^p h_{k,t-1} + \beta_{jk}^c h_{k,t}) \rightarrow (5)$$

$$\log\left(\frac{q_{jt}}{1 - q_{jt}}\right) = \beta_{j0}^p + \sum_{k \neq j} (\gamma_{jk}^p h_{k,t-1} + \gamma_{jk}^c h_{k,t}) \rightarrow (6)$$

Eq. (5) and (6) hold on the windows  $\{t : h_{k,t-1} < 0.5\}$  and  $\{t : h_{k,t-1} \geq 0.5\}$ , respectively. In the equations, the actual superscripts  $p$  as well as  $c$  indicate ‘previous’ as well as ‘current’ windows, respectively. Within this setup, each regression coefficient includes a straightforward perception. For case in point, the intercept term  $\beta_{j0}^p$  could be the baseline log odds with regard to  $0 \rightarrow 1$  transition in series  $j$  when all other correlated series are typically in the background state (state 0) with windows  $t - 1$  as well as  $t$ .  $\beta_{jk}^p$  and  $\beta_{jk}^c$  are the increase from the log odds of  $0 \rightarrow 1$  transition in series  $j$  when previous as well as current windows are usually in binding/modification state (state 1) in series  $k$  ( $k \neq j$ ), respectively.

With the current form, nevertheless, the quantity of regression coefficients can keep growing while incorporated more collection from the analysis. To treat this worry, a sparsity constraint is imposed using a LASSO penalty that is given in eq. (7).

$$\sum_{k \neq j} (|\beta_{jk}^p| + |\beta_{jk}^c|) \leq \lambda \text{ and } \sum_{k \neq j} (|\gamma_{jk}^p| + |\gamma_{jk}^c|) \leq \rho \rightarrow (7)$$

The details of this estimation process, including this coordinate lineage algorithm, are presented in the Supplementary Information. Finally, our major application of interest is read count data by ChIP-seq experiments, so a versatile class of distributions used for emission, including zero-inflated mixture model for the background sites (state 0) and also generalized Poisson distribution for the binding or maybe modification sites (state 1). The computational time of AHMM is even less than which of HMM. To view this, note that there are two elements in the entire computation time of AHMM: one concerning inferring hidden states as well as the other regards learning the regression coefficients. For the previous, the complexity is no more than  $O(N \cdot (4\alpha)T)$ , where  $T$  may be the additional time for computing the odds using  $(2N - 1)$  hidden state predictors from each transition, which needs to be trivial unless  $N$  is moderately large (should be all-around 1). For the latter, the computational complexity may the time it requires to outfit  $2N$  logistic regression models with LASSO penalty in datasets combine with  $(2N - 1)$  covariates and also  $T$  data items, and thus this is

time-consuming inside large datasets (large  $T$ ) just like genome-wide ChIP-seq data, and so at random, we sample genomic regions of a enough size and fit this model with all the subset in order to save time. Thus, the computation wants to be tremendously efficient than HMM.

## 2.2. Correlated big data Analysis

Execute a Correlated analysis to create the correlated degree matrix  $\Delta$  for a correlated dataset. This occurs in various ways with regards to the background information from the curator or maybe the characteristics from the dataset. Correlated analysis can even be carried out with no direct backdrop information. In this paper we tried the canonical correlation analysis (CCA) for finding the correlated matrix associated with Correlated big data. Canonical Connection Analysis is just about the generally used means of calculating the correlation between reduction techniques. Within CCA, two dissimilar representations given on the same pair of objects, and computed some sort of projection per representation such that they are maximally correlated to search for the correlation matrix. Technically, CCA computes two projection vectors  $\omega_x \in \mathbb{R}^d$ , as well as  $\omega_y \in \mathbb{R}^k$ , such that the correlation coefficient is defined in eqn. (8) gets maximized.

$$\Delta = \frac{\omega_x^T XY^T \omega_y}{\sqrt{(\omega_x^T XY^T \omega_x)(\omega_y^T XY^T \omega_y)}} \rightarrow (8)$$

Since  $\Delta$  is invariant to the scaling of  $\omega_x$  and  $\omega_y$ , CCA may be formulated equivalently as

$$\max_{\omega_x, \omega_y} \omega_x^T XY^T \omega_y \rightarrow (9)$$

Where;  $\omega_x^T XY^T \omega_x = 1$  and  $\omega_y^T XY^T \omega_y = 1$ .

### a. Correlated Sensitivity

We first analyze the source of unwanted noise on the global level of sensitivity prior to proposing the real correlated sensitivity. Traditional global sensitivity results in redundant noise produced by both Records and queries. As evaluated earlier, the regular method takes on Records generally are completely correlated with each other, and thus, it just multiplies the real global sensitivity using the maximal volume of Correlated big data resulting in large noise. Without taking into consideration the property of different queries for a new query, the standard method runs on the fixed global sensitive. In right fact, only a lot of the responding Records are correlated with other people, and we must consider the actual correlated information within people Records. Therefore, sensitivity ought to be adaptive for both Correlated big data along with the query.

Dependent on this observation, we first begin the notion of document sensitivity related with the correlated degree of each record, and then propose this correlated sensitivity of this query.

**Records Sensitivity:** For a given  $\Delta$  and a query  $Q$ , the Records sensitivity of  $r_i$  is given in eq. (10).

$$CS_i = \sum_{j=0}^n |\delta_{ij}| \left( \|Q(D^j) - Q(D^{-j})\|_1 \right) \rightarrow (10)$$

Where,  $\delta_{ij} \in \Delta$ . The Records sensitivity measures the effect on most Records with  $D$  on the deletion of a Record  $r_i$ . Estimate the correlated degree in between Records  $r_i$  and  $r_j \in D$  from  $\delta_{ij} \in \Delta$ . This notion combines the number of Correlated big data and the correlated degree together. If  $D$  is an independent dataset,  $CS_i$  is comparable to the global sensitivity.

**Correlated Sensitivity:** For a query  $Q$ , determined the correlated sensitivity by the maximal Records sensitivity.

$$CS_q = \max_{i \in q} (CS_i) \rightarrow (11)$$

Where  $q$  is a Records set of all Records giving an answer to a query  $Q$ . Correlated sensitivity related to query  $Q$ . It catalogs each one of the Records  $q$  giving an answer to  $Q$  and chooses the maximal Records sensitivity as correlated sensitivity. When any query simply wraps the independent or perhaps weak Correlated big data, the correlated sensitivity does not bring added noise.

Immediately after defining this correlated sensitivity for  $Q$ , the noisy answer is ultimately calibrated by the eq. (12).

$$\hat{Q}(D) = Q(D) + \text{Laplace}\left(\frac{CS_q}{\epsilon}\right) \rightarrow (12)$$

The observed correlated sensitivity is typically lesser versus global sensitivity, which assumes that every Records is usually fully correlated together and ignored this correlated degree.

### 2.3. Correlated Iteration Mechanism (CIM)

Although the correlated sensitivity brings upon a lesser quantity of noise as opposed to global sensitivity, when coping with many queries, the answers still get high noise considering that the privacy budget has e divided into several little parts. If the Records are in general strongly correlated with other folks, the noise is considerably advanced than the unbiased dataset. To undertake the issue, iterative-based mechanisms are going to be adopted for restricting the real noise within the query result. The major advantage from this mechanism is it can help save the comfort budget in addition to decrease the actual noise any time confronting a lot of queries. As a result, it is going to be appropriate pertaining to data releasing within the correlated dataset. On this section, we use a Correlated Version Mechanism (CIM) for answering a few queries within the correlated dataset. At first, we present a summary of the actual algorithm after which provides information on its operations.

This CIM plans to push out the results of a set of queries through iteratively updating the dataset. In this procedure, a dataset can be represented with a histogram  $x$  with length  $N$ . Let  $t$  are classified as the round index as well as the histogram being represented by  $x_t$  at the end of round  $t$ . We get a query set  $Q$  and choose a  $Q$  in each round  $t$ . We indicate the true answer as  $a_t$ , and the noisy answer as  $\hat{a}_t$ :

$$a_t = Q_t(x) \rightarrow (13)$$

$$\hat{a}_t = Q_t(x) + \text{Laplace}\left(\frac{CS_{qt}}{\epsilon}\right) \rightarrow (14)$$

Use this to control the updating round. CIM retains a collection of histograms  $x_1, x_2, \dots, x_t$ , which presents increasing approximation towards original dataset  $x$ . Use  $\hat{d}_t$  for denoting the difference between the correct answer from  $x_{t-1}$  and the noisy reply from  $x_t$ :

$$\hat{d}_t = Q_t(x_{t-1}) - \hat{a}_t \rightarrow (15)$$

This is utilized to control the updation round. CIM maintains a collection of histograms  $x_1, x_2, \dots, x_t$ , which represents an increasing approximation towards original dataset  $x$ .

#### a. Correlated Update Function

This part describes a correlated update function  $U$ . To get a histogram  $x_{t-1}$ , the actual function  $U$  recognizes all reacting Records  $r \in q_t$ . Per Records in  $q_t$ , all Correlated big data usually are denoted as a superset  $q_t$ . The update function  $U$  after that identifies some bins  $b$  that includes  $q_t$  and re-arranges the actual frequency of every bin in  $b$ . The final frequency on the  $x_t$  will become normalized so they sum to 1.

Let  $x_0, x_1, \dots, x_t$  be a histogram sequence, the function  $U$  is defined as a correlated update function if it satisfies the condition of  $x_t = U(x_{t-1})$ . The function  $U$  can be expressed as given in eq. (16).

$$x_t(b_i) = x_{t-1}(b_i) \cdot \exp(-\eta \cdot \delta_{qt} \cdot y_t(x_{t-1})) \rightarrow (16)$$

Where  $y_t(x_{t-1}) = Q_t(x_{t-1})$  if  $\hat{d}_t > 0$  and otherwise,  $y_t(x_{t-1}) = 1 - Q_t(x_{t-1})$ . 'η' is an update parameter associated with the number of maximal update rounds. The exacting correlated update function is dependent on the intuition that in case the answer produced by  $x_{t-1}$  is as well small in contrast to the genuine answer, the frequency of the relative bins will be enhanced. Usually, we may decrease the frequency if the answer is too large.

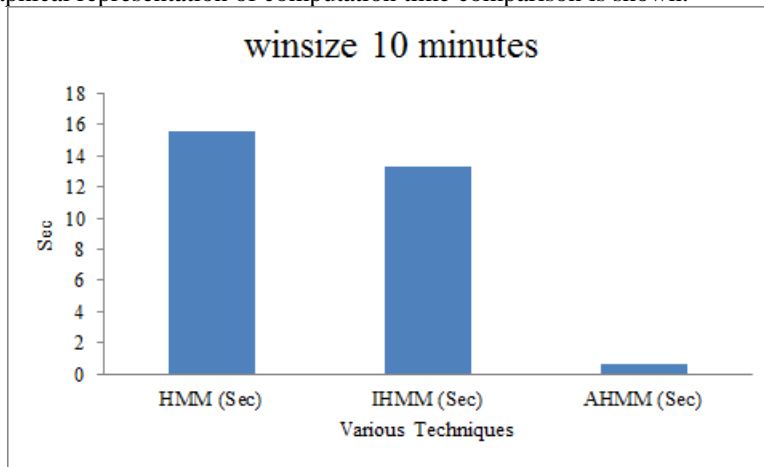
### 3. IMPLEMENTATION RESULTS AND DISCUSSION

The proposed system implemented for the privacy preserving of correlated dataset using JAVA language. Identification of Correlated big data, analysis of Correlated big data, and Correlated Iteration Mechanism are the three stages of the proposed system for execution. Consider the datasets in this paper are Adult, IDS and NLTCS. Then analyzed the performance of the proposed method using the comparison with conventional method. To reduce the computation time is an important reason for using the Adaptive HMM and it is proved by the comparison with HMM, IHMM and AHMM.

**Table 1:** Computation time comparison

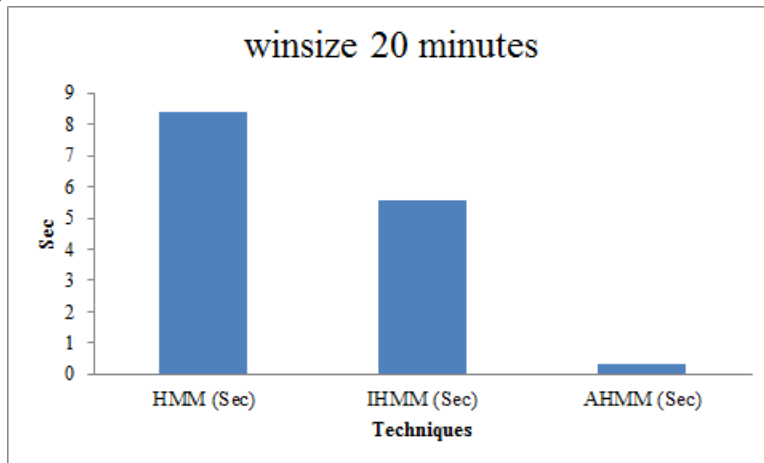
Winsize (minutes)	HMM (Sec)	IHMM (Sec)	AHMM (Sec)
10	15.564	13.274	0.63
20	8.419	5.584	0.303
30	6.548	4.568	0.248
60	3.21	2.224	0.145

In fig 2 to 5 the graphical representation of computation time comparison is shown.



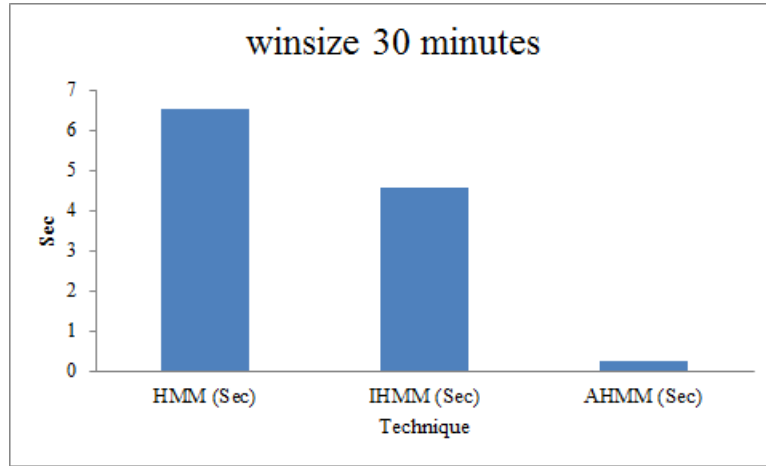
**Figure 2:** Computation time comparison at winsize 10 minutes

Fig 2 shows that the computation times of various techniques at 10 minutes winsize. The computation time of proposed AHMM is only 0.635seconds; however other techniques took 13.274 seconds by IHMM and 15.564 seconds by HMM.



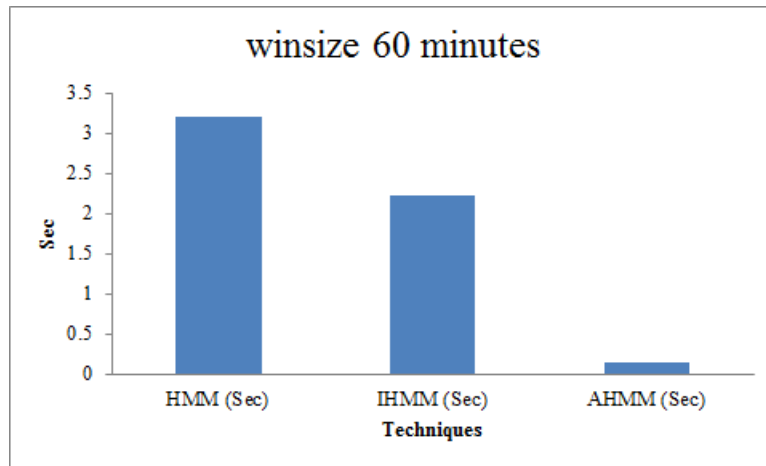
**Figure 3:** Computation time comparison at winsize 20 minutes

Fig 3 shows that the comparison analysis on computation time of various techniques at 20 minutes winsize. The computation time of proposed AHMM is only 0.303seconds; however other techniques took 5.584seconds by IHMM and 8.419seconds by HMM.



**Figure 4:** Computation time comparison at winsize 30 minutes

Fig 4 shows the comparison of computation time of various techniques at 30 minutes winsize. The computation time of proposed AHMM is only 0.248sec; however other techniques took 4.568sec by IHMM and 6.548sec by HMM.



**Figure 5:** Computation time comparison at winsize 60 minutes

Fig 5 shows the comparison of computation time of various techniques at 30 minutes winsize. The computation times of various techniques are 0.145seconds, 2.224seconds and 3.21 seconds by AHMM, IHMM and HMM respectively. Comparison of correlated sensitivity of proposed CCA with convention methods like global sensitivity (GS), correlation sensitivity (CS) and correlation sensitivity based on CCA (CS-CCA) is given in fig 6-8. Plotted the correlation sensitivity plotted using Mean Absolute Error (MAE) Vs differential privacy.

Table 2: Comparison of correlated sensitivity

Differential Privacy	MAE								
	Adult			IDS			NLTCs		
	GS	CS	CS-CCA	GS	CS	CS-CCA	GS	CS	CS-CCA
0.1	2.4	1.5	1.2	5.4	2.2	1.5	7	3	1
0.2	1.25	0.75	0.6	2.5	1	0.5	3.5	1.5	0.8
0.3	0.75	0.5	0.5	1.8	0.7	0.45	2.5	1	0.7
0.4	0.6	0.4	0.3	1.3	0.5	0.4	1.8	0.9	0.6
0.5	0.5	0.3	0.25	1	0.45	0.35	1.5	0.8	0.5
0.6	0.4	0.2	0.16	0.9	0.4	0.3	1.2	0.7	0.4
0.7	0.3	0.16	0.1	0.8	0.35	0.25	1.1	0.6	0.3
0.8	0.2	0.13	0.08	0.7	0.3	0.2	1	0.5	0.2
0.9	0.15	0.1	0.06	0.6	0.25	0.15	0.9	0.4	0.1
1	0.1	0.08	0.04	0.5	0.2	0.1	0.8	0.3	0.09



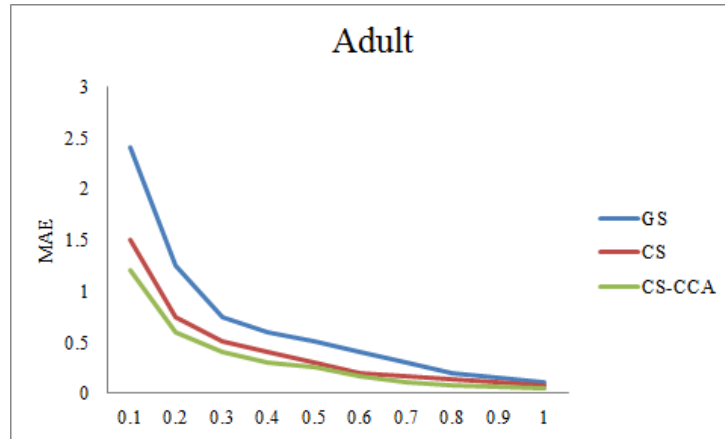


Figure 6: Comparison of correlated sensitivity for adult dataset

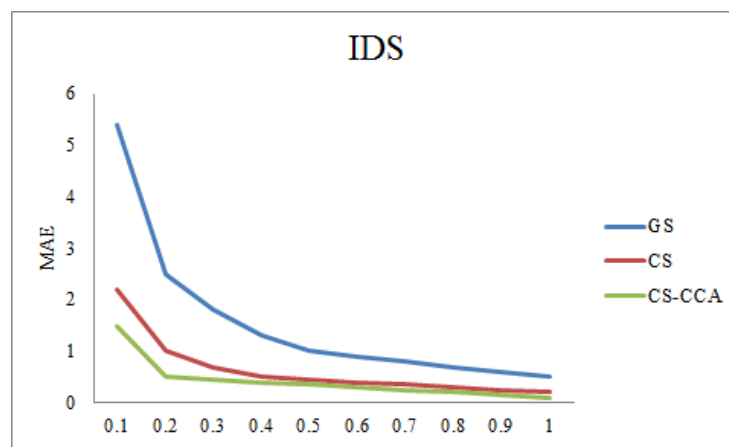


Figure 7: Comparison of correlated sensitivity for IDS dataset

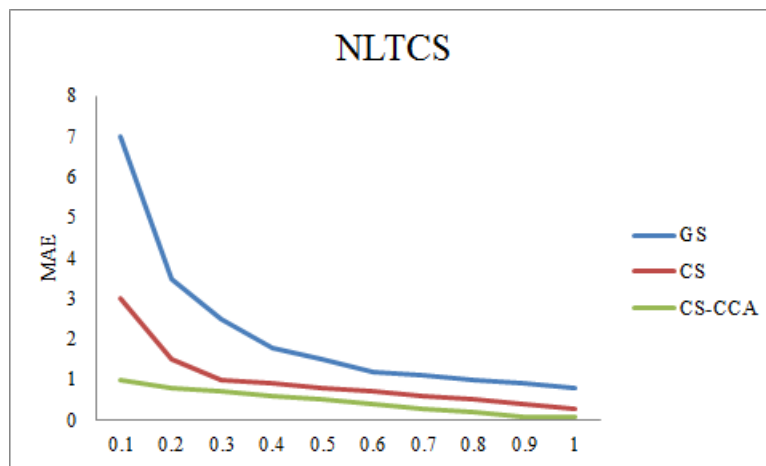


Figure 8: Comparison of correlated sensitivity for NLTCES dataset

Figure 6 to 8 shows the correlated sensitivity of the various techniques at dissimilar datasets, for all dataset the correlated sensitivity of the proposed CS-CCA method has better performance than the conventional GS and CS methods. So, for performances analysis proved that the performance of the proposed method is comparatively better than that of the conventional methods. Hence, we suggest that the proposed system can become a better option for the privacy preserving of correlated dataset.



## 5. CONCLUSION

The methodology for preserving the privacy of Correlated big data using the techniques like adaptive HMM and CCA is proposed. The proposed system is executed in three phases, in the first phase identified the correlated information from the Correlated big data and secondly analyzed the Correlated big data to calculate the correlated matrix and at last designed the correlated iteration mechanism to answer the queries. An adaptive hidden Markov model is proposed for the identification of Correlated big data, and then applied the CCA for Correlated big data analysis. The proposed method implemented using JAVA and analyzed the performance. The performance analysis suggests that the proposed method become a better option for preserving the privacy of Correlated big data.

## REFERENCE

- [1] Ibrahim AbakerTargio Hashem, Ibrar Yaqoob, Nor BadrulAnuar, Salimah Mokhtar, Abdullah Gani, and Samee Ullah Khan. "The rise of big data on cloud computing: Review and open research issues", *Information Systems*, Vol. 47, pp. 98-115, 2015.
- [2] CL Philip Chen, and Chun-Yang Zhang, "Data-intensive applications, challenges, techniques and technologies: A survey on Big Data", *Information Sciences*, Vol. 275, pp. 314-347, 2014.
- [3] Umarani and Punithavalli, "A Study on Effective Mining of Association Rules from Huge Databases", *International Journal of Computer Science and Research*, Vol. 1, No. 1, pp. 30-34, 2010.
- [4] Punam V. Khandar and Sugandha V. Dani, "Knowledge Discovery and Sampling Techniques with Data Mining for Identifying Trends in Data Sets", *International Journal on Computer Science and Engineering*, pp. 7-11, 2011.
- [5] Guo Haixiang, Li Yijing, Jennifer Shang, Gu Mingyun, Huang Yuanyue, and Gong Bing. "Learning from class-imbalanced data: Review of methods and applications", *Expert Systems with Applications*, Vol. 73, pp. 220-239, 2017.
- [6] S. Karthick, "Semi Supervised Hierarchy Forest Clustering and KNN Based Metric Learning Technique for Machine Learning System", *Journal of Advanced Research in Dynamical and Control Systems*, Vol. 9, pp. 2679-2690, 2017.
- [7] Carl Benedikt Frey, and Michael A. Osborne. "The future of employment: how susceptible are jobs to computerisation" *Technological forecasting and social change*, Vol. 114, pp. 254-280, 2017.
- [8] Liang Sun, Shuiwang Ji and Jieping Ye, "Canonical Correlation Analysis for Multilabel Classification: A Least-Squares Formulation, Extensions, and Analysis", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 33, No. 1, pp. 194-200, 2011.
- [9] Stephen Kaisler, Frank Armour, J. Alberto Espinosa, and William Money. "Big data: Issues and challenges moving forward." In System sciences (HICSS), *46th Hawaii international conference on*, pp. 995-1004. IEEE, 2013.
- [10] Sushma, Neelam, and Gajjala Venkata Kondareddy. "REPUTATION-BASED TRUST EVALUATION BY EXTRACTING USER'S ASSESSMENT", *International Journal of Technical Research and Applications*, Vol. 4, No. 1, pp. 5-16, 2016.
- [11] Tamir Tassa, "Secure Mining of Association Rules in Horizontally Distributed Databases", *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 4, pp. 970-983, 2014.
- [12] CL Philip Chen, and Chun-Yang Zhang. "Data-intensive applications, challenges, techniques and technologies: A survey on Big Data." *Information Sciences*, Vol. 275, pp. 314-347, 2014.
- [13] Anna Monreale, Salvatore Rinzivillo, Francesca Pratesi, Fosca Giannotti, and Dino Pedreschi. "Privacy-by-design in big data analytics and social mining", *EPJ Data Science*, Vol. 3, No. 1, pp. 10, 2014.
- [14] Wei Fang, XueZhi Wen, Yu Zheng, and Ming Zhou. "A survey of big data security and privacy preserving." *IETE Technical Review*, Vol. 34, no. 5, pp. 544-560, 2017.
- [15] Pingshui Wang, "Survey on Privacy Preserving Data Mining", *International Journal of Digital Content Technology and its Applications*, Vol. 4, No. 9, pp. 1-7, 2010.
- [16] Noman Mohammed, Alhadidi D, Fung B.C.M, Debbabi M, "Secure Two-Party Differentially Private Data Release for Vertically Partitioned Data", *IEEE Transactions on Dependable and Secure Computing*, vol. 11, no. 1, pp. 59-71, 2014.
- [17] PuiKuen Fong, Weber-Jahnke J.H, "Privacy Preserving Decision Tree Learning Using Unrealized Data Sets", *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 2, pp. 353-364, 2012.
- [18] Jaideep Vaidya, BasitShafiq, Wei Fan, Danish Mehmood and David Lorenzi, "A Random Decision Tree Framework for Privacy-preserving Data Mining", *IEEE Transactions on Dependable and Secure Computing*, Vol. 11, No. 5, pp. 399-411, 2014.
- [19] Xuyun Zhang, Laurence T. Yang, Chang Liu and Jinjun Chen, "A scalable two-phase top-down specialization approach for data anonymization using mapreduce on cloud", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 25, No. 2, pp. 363-373, 2014.

### BIOGRAPHIES OF AUTHORS

	<p>Sujatha Krishna received the B.E. and M. Tech degrees in Computer Science and Engineering from Visvesvaraya Technological University, Karnataka, India. She is currently pursuing the Ph.D. degree in the Computer Science and Engineering from REVA University, Karnataka, India. She is currently working as a Lecturer at University of Technology and Applied Sciences-Shinas, Sultanate of Oma. Her research interests include bigdata, data mining, machine learning and privacy preserving algorithms. She can be contacted at email: <a href="mailto:sujathasjit@gmail.com">sujathasjit@gmail.com</a></p>
	<p>Rajesh N completed his Ph.D. in Computer Science from Bharathiar University, Master of Computer Application from Thiruvalluvar University and BSc Computer science from Madras University. He is currently working as a Lecturer at University of Technology and Applied Sciences-Shinas, Sultanate of Oman. His research interest includes data mining, machine learning, big data analytics, blockchain technology, and Data Privacy and Security. He has published articles in the National and International Indexed journals, including SCI, WoS, SCOPUS.</p>