I J C T A, 9(28) 2016, pp. 343-356

AN ENHANCED ANONYMIZATION AND ACCESS CONTROL APPROACH FOR PRESERVING RELATIONAL DATA STREAMS

Kishore Verma. S* Rajesh. A and Adeline Johnsana. J.S*****

Abstract: The arena of privacy has realized a speedy developments in current years because of the growths in the capability to store and manipulate data. Certainly new improvements in the data mining field have led to increased concerns about privacy too. Privacy preserving data mining is a growing field of research which is focussed on the two contradictory goals (i) Utility and (ii) Privacy. Many privacy protection and access control approaches have been proposed for the continual relational data streams. Access control mechanisms for a relational data stream permits access to approve sliding window by just taking into account the consent utilized on each roles (role-based access control).Whereas Privacy protection mechanisms accomplishes privacy prerequisites by generalization of the stream data. While applying privacy protection generalization approach, which yields imprecision results can be reduced by delaying the publishing of stream data but in turn may prompts to have false-negatives. The challenge is to optimize the increase in imprecision due to the overlap of the rectangle bounds and also protecting the privacy. We formulated the problem as ACAC Accuracy Confined Access Control for preserving relational data stream, and given solution by incorporating an indexing mechanism of R*-tree to our X-TIM that effectively reduces the imprecision and produces the optimized results than previous TIM method.

1. INTRODUCTION

Now a days, a wide-range of applications handles data not as determined relations but rather as transitory data streams. For example applications such as financial sectors, networking, security provisioning, telecommunications, websites and sensor networks etc. are the celebrated fields which spend much of the time in managing and manipulating data streams. Extracting knowledge from these continuous data streams may aid field stake holders to derive the behaviour of the personalities, who are all involved in the knowledge domain. In doing so, there may be several privacy threats that may rise during extraction of knowledge from the continuous data streams. However the data mining on data streams are in need of mechanism that protects the streaming data from linkage attackers and also a mechanism that restricts the unauthorized access over the continuous data streams that are queried by the user / analyst. First, a well know existing data anonymization technique k-anonymity[1],[2],which protects the static data from linkage attacks have been revised in diverse perspective by [3] , [4] and [5] made it to applicable on streaming relational data. These mechanisms attempts to protect the relational data streams from linkage attacks. Second, access control approaches for determined relations concentrates in providing authorized admittance to the relations , that are need to be projected as resultant of user/analyst query processing. Whereas the case with respect to relational data streams, is little bit complex task due to the nature of the data i.e. they are dynamic and continuous. Access control on relational data streams strives to provide admittance only to the authorized parts of the data streams to each specific user/role, thus entity sheltered access control approaches are the views/ queries of the data streams. Uncertainly there are chances, which

* Research Scholar, Department of Computer Science Engineering, SCSVMV University, **Email:** kishore.saj3@gmail.com.
** Professor and Head, Department of Computer Science Engineering, C.Abdul Hakeem College of Engineering & Technology, **Email:** amrajesh73@gmail.com
*** Research Scholar, Department of Computer Science Engineering, St.Peter's University, **Email:** adeline.j.s@gmail.com

rules out the privacy of the sensitive attribute belonging to each record of the relation under authorized view. Furthermore, before revealing the relational data streams under authorized view of the admissible users/roles, privacy of the sensitive attributes need to be protected. Thus in order to provide authorized view over protected data streams there is need to blend the two widespread mechanisms called k-anonymization over relational data streams and access control approach. Under data anonymization perspective several approaches have been proposed so far to protect the static data from linkage attacks and then under authorized view perspective too several approaches have evolved. Mostly up to our knowledge only very few methods strives to provide authorized view over the protected data streams, recently [6] have been proposed, which consists two main concentration of task execution 1) Privacy Protection Mechanism and 2) Access Control Mechanisms, here the authors blends the two widespread concepts and strived to provide authorized view over the protected data. Thus initially the relational data stream records, which are selected as account of user queries are anonymized using k-anonymization strategy before letting it to the user protected view and meanwhile the access control mechanism employed in this blend will restrict the unauthorized user query generation and access over the records that are attained as account of querying. However, under privacy protection mechanism, anonymization the relational data streams through k-anonymity generalization lead to have imprecision, which can be minimized by dragging the publication of anonymized data stream records. But the counter fact issue is, when they attempt to delay the publication of anonymized data streams on account to reducing the imprecision, lead to have more no of false positives and false negatives. Due to the increase in the false-positives and false negative, increases the query limit violation which subject to increase the average query limit violation. Thus under these scenarios if authorized user generates query, he is unable to view his intended records, so the utility of systems is dropping out.

Our proposal is the extension of the recent previous work [6], that we strived to employ mishmash of novel data stream anonymization algorithm and X-TIM (Extended Total Imprecision Minimization) approach, which provides the restricted view over the anonymized records that's being satisfied by user generated query under privilege levied to each individual user / role. Essentially our approach is capable of anonymizing the relational data stream with minimized imprecision which lead to have less information loss and also prevents them from linkage attacks. The core element of our proposal is the X-TIM, as justified part we incorporated R* tree indexing [7] approach in handling the access control operation, since R* tree is a proficient indexing strategy energizes storage utilization with little expense in complexity, our approach avoids the numerous duplications of the records that need to be published as a anonymized view of relational data streams under admitted access. Thus this elimination of duplicate records, contained in the leaf node allows still more records to participate for privacy preserving data stream publication, which lead us attain less number of false positive and false negatives records. This minimized false positives and false negatives attained through our approach have made our methodology as promising one in consuming less average query limit violation than the other existing methodologies. The ultimate aim of our proposed work is to minimize imprecision caused due to k-anonymity generalization by introducing an enhanced anonymization algorithm which causes less imprecision than[6] and revised version TIM algorithm using R* tree to reduce the average query violation limit, , the factor that decides the hit ratio of tuples that's being satisfied towards the user generated queries.

The remainder of this paper is organised as follows Section 2.Litrature survey – we discuss the various methodologies that relates our proposal, Section 3. Background – here we had shown the basic initiatives need to be known in understanding our flow of approach, Section 4. A detailed discussion about our novel proposed work, Section 5. We discuss the experimental setup, issues and outcomes of our approach and Section 6. We conclude the effectiveness and efficiency of our system and suggested future directions.

2. LITERATURE SURVEY

Here we discuss the various well known mechanisms that given way to formulate our novel strategy ACAC methodology in anonymizing the relational data streams and viewed under privileged rights. A recognized approach to anonymize data is the K-anonymity [1], [2], which contributes in preventing data from identity disclosure and linkage attacks. A relational table RT adheres k-anonymization by distorting its QI (quasi identifiers) attributes, such that each generalized QI set appear k- times in the anonymized relational table. Condensation [8] [9] are the equivalent method of k-anonymization have been proposed, which divides the relational data table into fixed size group appealing k-anonymization principles. Even though k-anonymization [1], [2] suffers from background knowledge attack and homogeneity attacks, this approach is well enabled to protect the records from identity disclosure. L-diversity [10] overcomes the problems of background knowledge attack and homogeneity attacks that seems to weaken the k-anonymization. This method ensures that there may be at least l-sensitive attributes values present in each k-groups, however in doing so, at some perspective of application l-diversity give way to reduce the sematic relationship between the records and attributes(i.e. increases the semantic loss. T-closeness [11] derived and proposed a solution to maintain the semantic relationships in the relational table, which usually get drip out on applying k-anonymization and its successor's l-diversity [10].

The traditional k-anonymization strategies are well enough methodologies to handle static relational data, but didn't not gain feasibility in handling streaming data. [3] SKY is the first proposal delivered to anonymize the streaming relational data and continuously maintains the δ constraint in anonymization strategy to increase the utility over the anonymized data streams but fails in handling voluminous data arrivals. [4], [5] anonymizes the streaming data tuples with convincing loss, especially [5] practises min-delay strategy to furnish anonymization process with minimum loss, in some extent which may lead to have more number of false positives. Thus there exist mismatch in maintaining the min-delay constraint on each views, which may lead to have unrealistic modulation in calculating the false positives and false negatives on each views. [12] Conveys the greater insights of feasible and non-feasible strategy residing in handling views of the relational data streams, whereas [13] proposed a solution that anonymize the data streams in the generated views with minimum delay constraint. Under mysterious views generation [13] grieves in handling the false negatives in terrible dithering stage. [14] is the combined scheme of handling streaming data under access control mechanism, attempts to achieve k-anonymization under distributed client/server setup, which is not our part of focus.[15] proposed few significant standards need to be carried out in achieving a better utility and security under role based access control's prime motivation. [16] Extends the role base access mechanisms to deliver fine grained admittance towards the tuples that's being queried by the user. [17] privileges the multi user roles can have their authorized views over the tuples simultaneously.[18] extends the access control mechanism by a hybrid combination which explores, that a group of authorized predicates is being restricted for individual user, under anonymized data.[19] proposed access control mechanisms to manage personal health records under semi honest platform. None of the above methods supports access control over anonymized , accordingly[17] schemes that relations under access control mechanism over data will not guarantee the privacy of sensitive attributes present under authorized view. [20] Merges the privacy protection mechanism with access control policies to protect the sensitive attributes to the authorized user, under static environment. [6] extends [20] approach to handle streaming data, which supports access control over protected data streams by combining k-anonymization of sliding window data streams and access control mechanism to encourage the authorized view over anonymized data streams. However anonymization of data streams through k-anonymity generalization reveals imprecision, which is minimized by delaying the process of publishing streaming data subject to anonymization, this may lead to have more number of false negatives that increases the average query limit violation, the factor that decides the hit ratio of tuples that's being satisfied towards the user generated queries.

3. BACK GROUND

Let streaming data record Rs[i] = {Id, ts, At1, At2, …Atm, Atj, Ats}, Where Id denotes the identity attribute, ts is the time stamp variable denoting the arrival time instance of the streaming data record, Aj is the quasi identifier attribute, Atm is the number of quasi identifiers arrived, Ats is the sensitive attribute. Rs[i] characterises all the streaming record that have reached till time instance I, Id acts as the identity (e.g. ssn) that uniquely identifies an individual in a group of streaming data. (At1 to Atj) are the quasi identifiers which can be used in linkage attacks to infer privacy informations.

2.1 Definition (Data stream k-Anonymization) [3]

Let Rs[i] = {Id, ts, At1, At2, …, Atj, Ats} be a data stream record, where {At1, At2,…, Atj} are the quasi identifiers, Id is the individuals identity, ts is the arrival time stamp and At1, At2,….., Atj are the remaining data stream attribute. Let be the anomnymized version of streaming data generated from where Id have been cropped. We say that is k-anonymized, when the following conditions are satisified,

- For each record r $\in$ R_s, there exists a anonymized record in R_s^*
- Given a record $\overline{r} \in R_s^*$, we formulate qs(quasi set) as corresponmding QI set, where $qs = \{\overline{r'} \in R_s^* \mid \overline{r'},\ q_j,\ j \in j\ [1,...n]\}$

Given a QI set *qs*, $IR_{(qs)}$ let be the set of individual records of the persons belongs to *qs*. For each possible individual $qs \subset R_s^*, | \geq k$

Here we denote *k*-anonymity over data streams is their newness.

Definition 2 Delay Constraint [6]

Let S be a -anonymization scheme receives input data stream and produces generalized output data stream and let β be the non-negative integer. We state that S adheres the delay constraint β if and only if each newly arrived record r ∈ R_s with its time stamp more than all the arrived record's time stamp. The definition explains when a new record $\overline{r}$ arrives, the record should be with a time stamp greater than all other arrived records.

Definition 3 Equivalence Class [6]

An equivalence class is formed by framing set of records having the same Atj QI attribute and ts time stamp value.

Definition 4 Role Base Access Control:

A RBAC strategy ρ is a record {IU,R,A,UR,HR}, where IU is group of individual users, R is collection of roles , UR is the user-role mapping, A is authorizations, HR is the Hierarchy of Roles, RA is a role-authorization mapping etc. Role base access control for streaming data delivers a sliding window predicate that states authorized data stream view.

4. PROPOSED WORK

Here we explain the Imprecision, Imprecision bounds and average query bound violation (AQV).

Sliding – Window Query assessment and Imprecision: A sliding window query is exercised for the streaming data records R_s[i]all data stream record R_s[i] that adheres the query all included , means all the records that are being overlapped due to anonymization are also includes.

Definition (False- Positive record) [6]

A record is said to be false positive , when it does not actually satisfies the query , but being included in the result based on the overlap introduced an account of *** equivalence in T[i] that contains the overlap tuples.

The count of false positive records in the outcome of a sliding window query $SWQ_j[i]$ at any instance I of time is derived as follows

$$FP_{SWQj}[i] = |_{SWQj}(R_s^{\#}[i])| - |_{SWQj}(R_s[i]) - (R_s^{h}[i])|$$

where $$|_{SWQj}(R_s^{\#}[i])| = \sum_{EC(overlaps)SWQj}|EC|.$$

The false positive records may be included in the published portion owing to a spatial overlap of Quasi Identifier attributes, time stamp overlap or both.

Definition (False- Negative record) [6]

A record is said to be false negative record when it contents the sliding window query at the query assessment instance, but not accounted in the query outcome due to being accommodated under hold.

The count of false negative record is the outcome of a sliding window query, $SWQ_j[i]$ at any instance i of time is derived as follows

$$FN_{SWQj}[i] = |_{SWQj}(R_s^{h}[i])|$$

If delay constraint is released, the number of false positive records is minimized because less imprecision records are formed under each equivalence class.

Definition (Query Imprecision of sliding Window) [6]

Query imprecision defines the total no of false positive and false negative records assessed under each query assessment over the anonymized data stream $T^{\#}[i]$ at given instance *i*.

Here sliding window $SWQ_j[i]$ is processed over $R_s^{\#}[i]$, which includes all the records of the equivalence classes that overlaps the query region.

Definition (Query Impression limit) [6]

It is defined as the total imprecision affordable to the access control mechanism, when a sliding window query $SWQ_j[i]$ is process at time instance *i*.

A query usually violates the imprecision bound (system parameters, only when total imprecision bound is greater than the predefined imprecision bound.)

Definition (Average query limit violation (AQV)) [6]

It is defined as rate at which the query imprecision bound is isolated for a given period of time $AQV_{Qj} = VQj / NQj$ where NQj is the number of steps taken to execute at time instance(i) and VQj is the number of time the imprecision bound been violates through these steps.

Definition 10 Expected False-Positives [6]: It is defined as the sum of all false-positives for a leaf node of Partition P at the current time instance.

$$EFPP = \sum_{Qj \in Q})|P - Q_j|$$

Definition 11 Expected False-Negatives [6]: It is defined as the sum of the false-negatives for all the queries which is to be executed at the next time instance from partition P if the partition is held by the PPM at the current time instance.

4.2 The ACAC Approach

Definition

Given a streaming data record $R_s[i]$, a set of sliding window queries q and privacy denotes K_s, the accuracy constraint access control mechanism strives to produce and anonymized stream $R_s^{\#}$, is being the sum of average query bound violation for entire queries q€ Q is reduced .

System Architecture

Accuracy Confined Access Control for privacy preserving relational data stream is proposed as shown in Figure 1 (System Architecture). The privacy protection mechanism ensures that the privacy and accuracy goals are happened before the sensitive streaming data made available to the access control mechanism. The access control administrator is responsible for providing the approved views of the data stream that contents the sliding window query (SWQ).The privacy preserving mechanism (PPM) ensures the privacy needs by using a novel DSA(Data Stream Anonymization) process ,which adopts generalization strategy to perform the anonymization task over the relational data streams. On account of applying generalization over the relational data streams, imprecision are generated, thus it can be reduced by extending the time of publishing the relational data stream. However these extension of time reads to have false positive records if some records are locked by privacy preserving mechanism. These imprecision minimization is meticulously done by optimized Imprecision minimization strategy (upgraded version of TIM [20]) using R* tree approach. The administrator (i.e.) system parameter defines the imprecision limit for each and every query process under them .Our approach carry over DSA a revised version of [3] to relational data streams, that protects linkage attacks being performed on individual records $R_s^{\#}$.

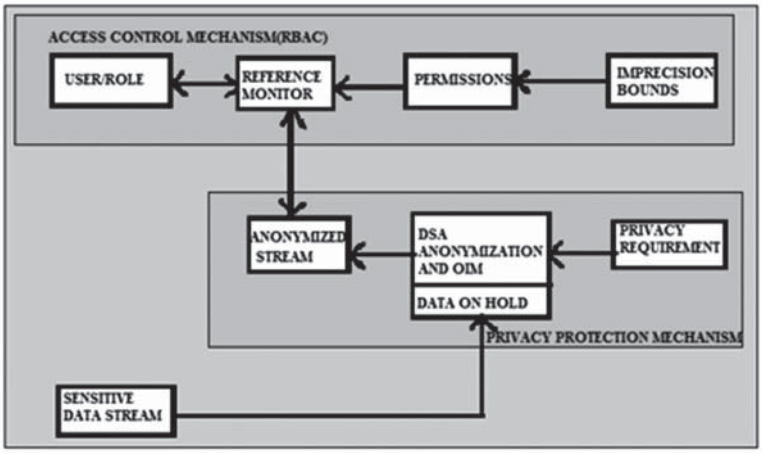

Figure 1. Accuracy Confined Access Control Framework

PPM ensures confirms that the sum of false negative and false positive records is smaller than the imprecision limit during the instance of query processing. Reference monitor play the role administrator task to set the type of semantic i.e. (i) either overlap semantics (ii) Enclosed semantics. False-positive are produced due to generalization overlapping semantics implies that access to the unauthorized records are provided implicitly .Then false negative is enclosed semantics, even though it denies access to authorized data but will not violate the access control policy. In our work we formulate and concentrate on derive mechanism that reduce the overlap by incorporating R* tree, so that the imprecision minimized under overlap semantics. Thus our approach attained less average query limit violation than the existing PACE approach.

4.3 Algorithms for Accuracy Confined Access Control Approach

[3] Have proposed stream k-anonymity which constantly apply k-anonymization on data streams, [5] extends k-anonymization to be applied enduring clustering approach. [4] proposed approaches

to do k-anonymization on data streams that arrives continuously view using R- tree [21] , traditional k-anonymization[1],[2] .The streaming data records are added to the leaf nodes of R^- tree indexing, by adhering a condition, that leaf nodes should accompany relational data stream records between k_s to $2k_s$.Once all the leaf nodes of the tree got filled by satisfying the condition k_s to $2k_s$, then that particular data stream records accompanied in the leaf nodes will be published as anonymized data streams. These published leaf nodes are subject to be removed from the R- tree indexing and again the same steps continues for the next batch of arrived relational data stream records. A recent work [6] applies the methodology of [4] but performs indexing using R+ tree through which [6] privileges to attain reduce imprecision.

Whereas our proposed methodology is based on two strategies

(i) We adapt the executional approach proposed by [6]

(ii) And replace the R+ tree indexing with that of R* tree [7] in order to resolve the greater imprecision generated due to overlapping of leaf nodes of relational data streams.

We state that R*- trees[7] are more compacted, efficient and accurate indexing structure than R+ trees, so which made us to incorporate the R* tree indexing to handle the arriving anonymized relational data stream records. Thus our novel approach had given way to reduce the imprecision raised due to anonymization of data streams, which guided us in attaining minimum average query limit violation than the existing approach [6].The increase in the imprecision of previous work Total Imprecision Minimization algorithm (TIM) is caused due to the repetition of data stream records in the leaf nodes inserted adopting R+ tree strategy. This increase in imprecision by default increases the chance of query violation generated by user (i.e. Increases Average Query Violation Limit).And also the previous work[6] executes only by considering the non-overlapping rectangles generated by anonymization process. Our proposed exploratory work approach listed in Algorithm 1(Data stream Anonymization) is applied on raw data streams to prevent the original data viewed under authorized from being exposed to linkage attacks.

Here in our approach we propose X-TIM (Extended Total Imprecision Minimization) that uses R* tree [7] indexing approach and strives to consider both overlapping and non-overlapping rectangles and executes to attain minimum imprecision and reduced average query limit violation. An inclusive of R* tree indexing in filling out the leaf nodes adhering the condition k_s to $2k_s$ reduces the duplication of data stream being filled in the leaf nodes. Thus this optimized characteristics of R* tree indexing helped us in achieving less number records accounted for being violated from AVQ(Average Query Violation Limit), which is significant factor in our proposals. Our work's X-TIM initializes the R* tree to handle the arriving anonymized data streams from the PPM , the relational data streams $R_s[i]$ are added to the leaf node of R* tree at each instance of time. Initially R* tree are made empty and the well-ordered data stream adhering the condition k_s to $2k_s$ are added to the leaf node N partition as per stated lines 1-8. If N is greater than M (Maximum allowed entries in the leaf node of R* tree). Then arrived data stream record records $R_s[i]$ need to undergo forced re-insertion, little bit expensive process. Thus the forced reinsertion is termed to be costlier R* tree allows only one reinsertion at a particular level, due to this, constraint strategy restructuring of tree nodes by forced reinsert is reduced and at the same time it improvises storage utilization. A prompt increase in the storage utilization eliminates duplicate records being placed in the leaf nodes of the R* tree used by X-TIM. Each time the anonymization range will be updated according to the properties of quasi-identifiers, and the data stream record , that are kept in hold for current time instance *i* would be tuned to get arrive into the anonymization process is done in line 13-16.

Algorithm 1. Data Stream Anonymization (DSA) algorithm

Input: a stream of records $R_s[i]$, set of ordered leaf nodes N, parameter k

Output: Equivalence classes EC1, EC2...with respect to set of partitions S.

Begin

1. Set the active R*-tree to an empty R*-tree
2. Initialization
3. While N not equal to empty.
4. P->empty partition
5. While |P|≤ M (where M represents max.no.of entries in leaf node)
6. L->next leaf node in N
7. Add all records in L to P
8. N->N-L
9. If the total number of records in the remaining leaf nodes in N is less than M then remove those records
   From N and add them to P
   Else
   For all the excess entries of the node N for the first call in the same level.
   Compute the distance between the centres of their rectangles and the centre of the bounding rectangle
   Sort the entries in decreasing order
   Remove the first entry and adjust the bounding rectangle.
   Else
   Remove the leaf node T, partition them into T1 and T2 and add them to P.
10. Update generalised quasi-identifier values for every record in P.
11. S→SUP
12. Continue step 3.
13. For each equivalence class EC that is due at the time instance i do
14. Examine and publish EC of the set of records to be held
15. Remove and Re-insert all those records held into R*-tree
16. End for

According to our proposed approach X-TIM listed Algorithm 2, first calls the DSA routine to anonymize the raw relational data streams. R* tree can have false positives (if Published) or false negatives (if held) executed towards sliding window queries. A false positives signifies the information loss obtained due to the generalization, whereas false negatives signifies the information loss obtained delay constraint followed in publishing the data streams. Therefore, we opt for key aspect that reduces imprecision levied on each partition projected for the queries generated, in other words an active R* tree holds the partition in the leaf node until EFP_p is smaller than EFN_p.

Algorithm 2 eXtended - Total Imprecision Minimization(X-TIM)

Input: $R_s[i]$, k_s, Q, and LQ_j

Output: EC1, EC2,

Begin

1. Call DSA algorithm.

2. If there is an overlapping of two rectangle choose the rectangle which needs least enlargement and Resolve the ties between them.
3. If (Size of new leaf nodes after splitting is > k_s) then
4. Split the leaf node;
5. For (all leaf nodes P in active R*-tree at time instant i) do
6. Update the imprecision cost of each leaf node;
7. If ((w_{FN} * EFN_P > EFP_P * w_{FP}) OR ((i – t_m.TS) (δ-1)))
   Then
8. Publish the leaf node as EC and remove from active R*-tree;

5. EXPERIMENTATION AND RESULTS

The hardware configuration used for the implementation is Intel(R) core TM i5-5200U CPU @2.20GHZ. The RAM capacity is 8.00GB. The Hard disk used for the implementation is 1 TB. It is a 64 bit operating system and the operating system used is Windows 7 ultimate. The tool which is used to run the execution is Microsoft visual studio 2010.The front end of our implementation is c#/.net and the back end used is Microsoft SQL server. The system is designed in such a way that any type of categorical dataset can be loaded and executed. We have considered the adult dataset for the execution with the attributes such as sex,age,race,marital-status,education,native-country,workclass,occupation,salary-class.

The records are first anonymized with respect to the value of k-anonymity and the stream of records is introduced in the time-sliding window for the duration of some time period. The records are bounded by the rectangles with respect to the two-dimensional quasi-identifiers. When the new record arrives it is kept in the minimum bounding rectangle under hold and it is not anonymized, they are arranged in the X-TIM R*-tree by considering the value of M which represents the maximum number of values that the minimum bounding rectangle can hold. If there is an overlap of record among the rectangles then the criteria of nearness is found and the records are pushed to the rectangle which requires the least enlargement. If the leaf node is filled and if a new record is to be inserted into that rectangle then the forced reinsertion concept of R*-tree is carried in which the distance from the centre to all the records are calculated and then it is arranged in the descending order and the maximum distance record is replaced with the new record and the replaced record will be placed in the neighbouring rectangle. Since it is very costlier, the process is carried only once and if the same situation arises again then the splitting takes place by arranging the records in two groups one in ascending order and another in descending order and splitted in equal halves if there is overlap then the nearness criteria is checked .Likewise the records are introduced in the R*-tree. The slide is now moved with the step of the time lesser than the duration period. The anonymized and published records of that time of the step will be deleted and the records which are arrived and put under hold are anonymized in the next time instance. Then the false-negatives and false-positives are calculated and then the imprecision is calculated by using it. The total number of imprecision is calculated and it is checked whether it violates the query bound and it is recorded whether it violates or not. This process is carried many number of times and average query bound violation is examined. The records are published to the user by calculating expected false- positives and expected false-negatives of the requested query by checking the permissions provided to the user.

The records are first anonymized with respect to the value of k-anonymity and the stream of records is introduced in the time-sliding window for the duration of some time period. The records are bounded by the rectangles with respect to the two-dimensional quasi-identifiers. When the new record arrives it

is kept in the minimum bounding rectangle under hold and it is not anonymized, they are arranged in the R*-tree by considering the value of M which represents the maximum number of values that the minimum bounding rectangle can hold. If there is an overlap of record among the rectangles then the criteria of nearness is found and the records are pushed to the rectangle which requires the least enlargement. If the leaf node is filled and if a new record is to be inserted into that rectangle then the forced reinsertion concept of R*-tree is carried in which the distance from the centre to all the records are calculated and then it is arranged in the descending order and the maximum distance record is replaced with the new record and the replaced record will be placed in the neighbouring rectangle. Since it is very costlier, the process is carried only once and if the same situation arises again then the splitting takes place by arranging the records in two groups one in ascending order and another in descending order and splitted in equal halves if there is overlap then the nearness criteria is checked .Likewise the records are introduced in the R*-tree. The slide is now moved with the step of the time lesser than the duration period. The anonymized and published records of that time of the step will be deleted and the records which are arrived and put under hold are anonymized in the next time instance. Then the false-negatives and false-positives are calculated and then the imprecision is calculated by using it. The total number of imprecision is calculated and it is checked whether it violates the query bound and it is recorded whether it violates or not. This process is carried many number of times and average query bound violation is examined. The records are published to the user by calculating expected false- positives and expected false-negatives of the requested query by checking the permissions provided to the user.

5.1 Loading the Dataset

The adult dataset is uploaded by clicking the load dataset in which the location of the database is tracked and loaded.

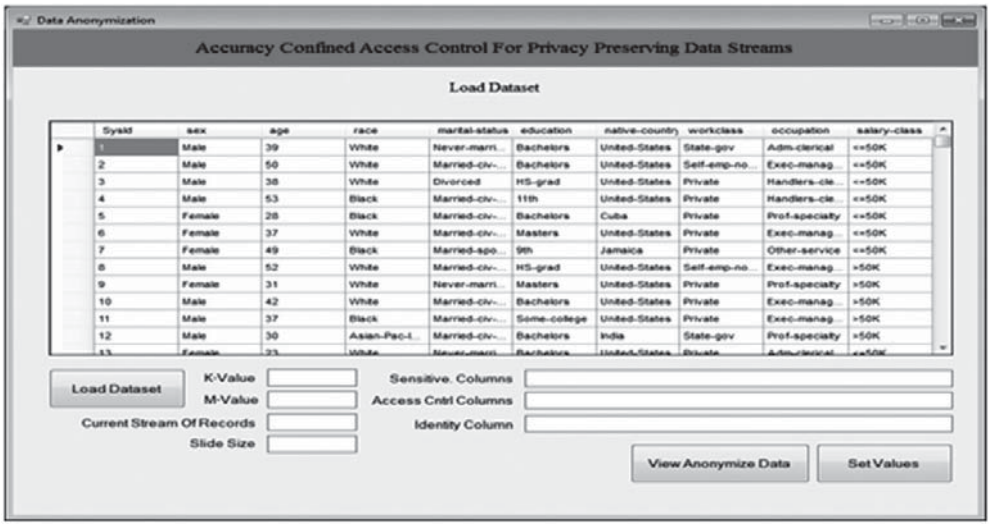

Figure 2. Loaded Dataset Display

5.2 Providing Permissions

Then the permissions are provided by the administrator by providing the values of K which is the anonymization value M, which is the value of maximum number of records that the rectangle bounds can hold. The sensitive column value, quasi-identifiers of the two dimensional columns are selected from the database uploaded and the number of records to be loaded at each instant is defined and then the sliding window value is set in such a way that it is lesser than the stream of records so that overlap occurs in the rectangle bounds and these permissions are set by clicking set values.

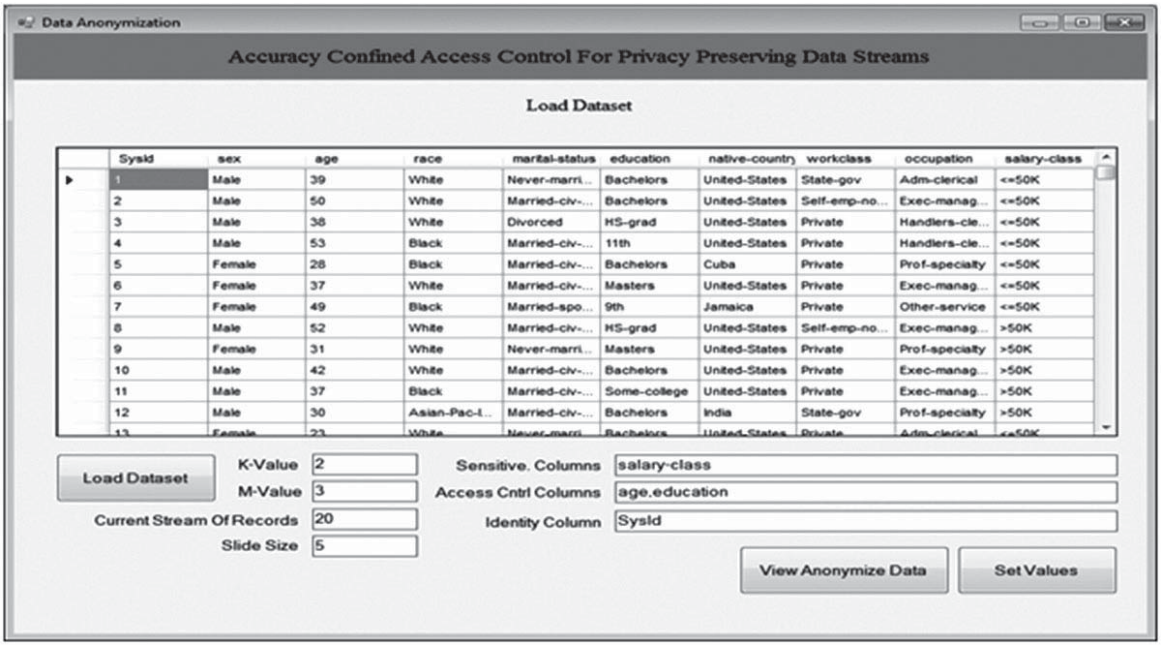

Figure 3 Providing Permission

5.3 Querying

Then the view anonymized value is clicked and a new window opens in which the previously set values are displayed. The dataset and the anonymized data are displayed for the stream of records at that instant. In the access control columns, the query regarding the quasi-identifiers to be published are provided by the user the query limit values are also set by the user.

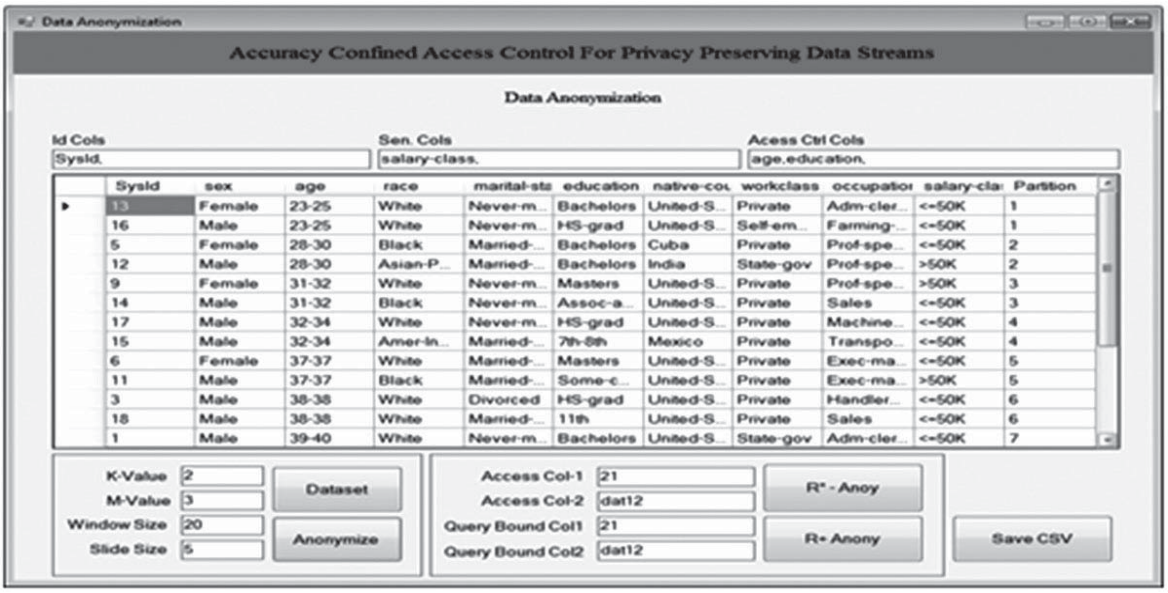

Figure 4 Query bound set and Query generation

The number of records in the slide are inserted and the same number of records is deleted in the existing records in both R*-tree and R+-tree and the current stream of records that are dealt at each time instant. The duplications and increase in imprecision is reduced and output is published according to the query.

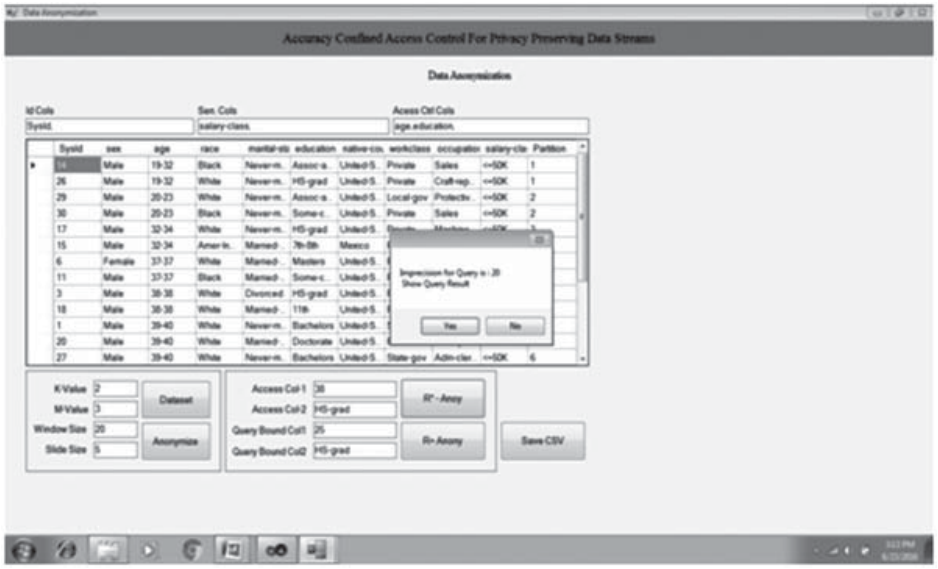

Figure 5. Imprecision Calculation for X-TIM

5.4 TIM

In R+-tree duplications of the identifiers arises and it increases the imprecision due to the duplication and the output is published to the user according to the query at the current time instance .Records are inserted and deleted according to the value of slide at each time instant.

The Average query limit violation under different query limit for same k value is examined for TIM and X-TIM algorithms respectively and it is found that the average query limit violation is considerably lower in the case of X-TIM's R*tree. So the proposed work R*-tree in X- TIM is better than the R+-tree in TIM.

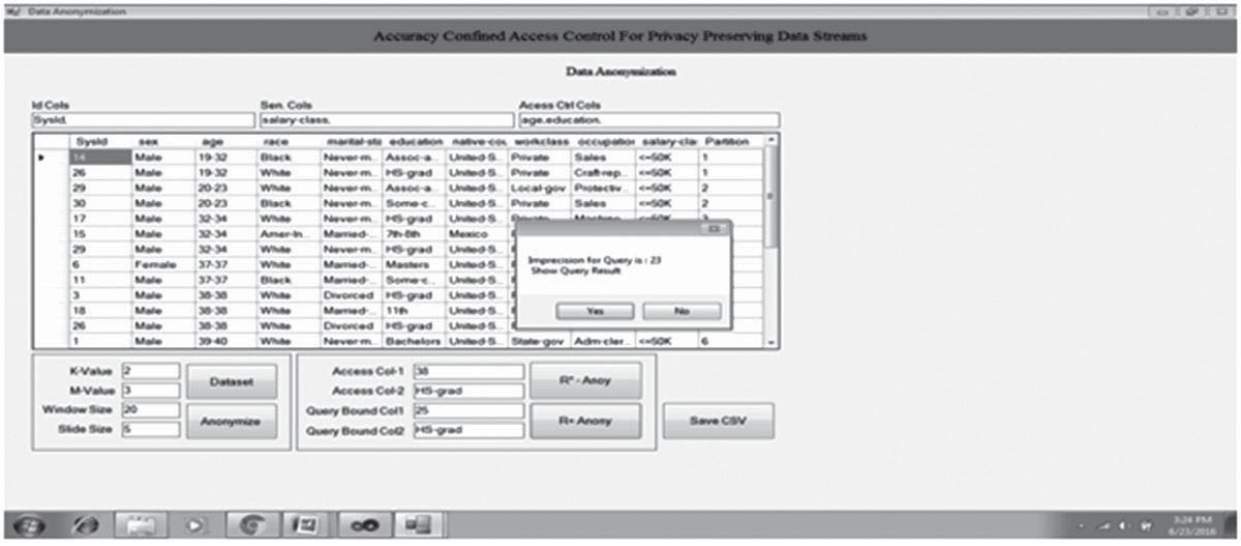

Figure 6 Imprecision Calculation for TIM

The Figure 7 represents comparative results of TIM and our approach X-TIM's information loss occurred due to k-anonymization process, from the experimentation it reveals that TIM has minimum information loss with respect to our approach, nevertheless it will be a convincing level of loss by considering the average query limit violation. Figure 8 and Figure 9 represents TIM's and our approach X-TIM' AQV values for the k values of 2 and 3 respectively and the experimentation confirms that our approach intends in producing reduced average query limit violation when compared to the previous TIM method.

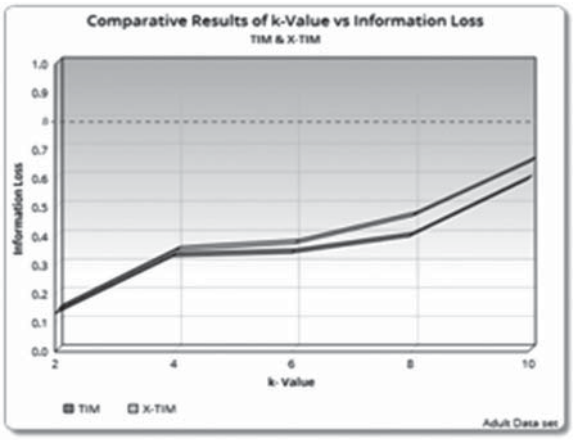

Figure 7. Information Loss comparison of TIM and X-TIM

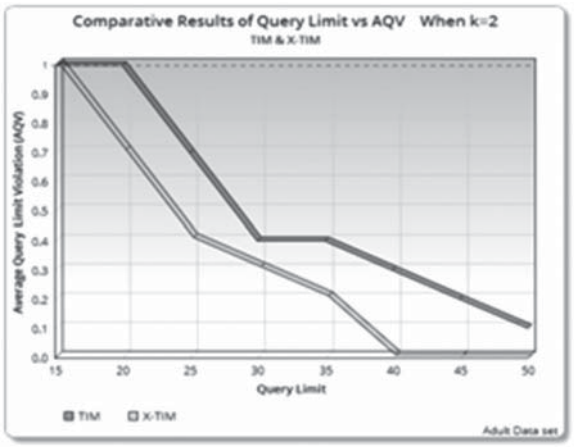

Figure 8. Average Query Limit Violation Comparison of TIM and X-TIM, k = 2

6. CONCLUSION

Protecting the relational data that is released for some beneficial purpose while preserving privacy of the individuals who own the data, our approach creates anonymization table from the original table by considering various constraints like accuracy constraints. Access control mechanism allows only authorized query predicates on the sensitive data. The privacy protection mechanism anonymizes the relation data to meet privacy requirements and imprecision constraints on predicates set by the access control mechanism and partitions are splitted by using Sliding-window. Access control mechanism for relational data is constructed with the privacy preservation based model. Accuracy Confined Access Control (ACAC) scheme provides security to the data by allowing access based on permissions assigned to the access. K-Anonymity model is integrated with our ACAC approach that leverages minimum imprecision based data access control mechanism. Partitioning using X-TIM R*-trees improves the imprecision minimization which reduced the violation of query limit and the average query violation is reduced .It is also useful in the case of storage utilization compared to TIM's R+-tree but it slightly cost higher than R+-tree.

Thus, the proposed system X-TIM protects the sensitive micro data with low imprecision and accuracy is improved.

There are many more interesting and important directions worth exploring. Though the proposed approach of R*-tree improves accuracy by reducing imprecision, it is slightly costlier than R+-tree. It is also of great interest to extend this approach to produce the accurate results with less cost.

References

1. P. Samarati and L. Sweeney, "Protecting Privacy When Disclosing Information: k-Anonymity and Its Enforcement through Generalization and Suppression," Technical Report SRI-CSL-98-04, 1998.
2. Latanya Sweeney, "*k*-Anonymity: A Model for Protecting Privacy" *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems,* 10 (5), 2002; 557-570.
3. C.C. Aggarwal and P.S. Yu, "A Condensation Approach to Privacy Preserving Data Mining," Proc. Ninth Int'l Conf. Extending Database Technology (EDBT), pp. 183-199, 2004.
4. V. Ciriani, S. De Capitani di Vimercati, S. Foresti, and P. Samarati, "*k*-Anonymity" © Springer US, Advances in Information Security (2007)
5. A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "l-Diversity: Privacy Beyond k-Anonymity," Proc. 22nd Int□l Conf. Data Eng. (ICDE), p. 24, 2006

6. N. Li, T. Li, and S. Venkatasubramanian, “t-Closeness: Privacy Beyond k-Anonymity and l-Diversity,” Proc. IEEE 23rd Int□l Conf. Data Eng. (ICDE), pp. 106-115, 2007.
7. Jianzhong Li, Beng Chin Ooi, Weiping Wang t “Anonymizing Streaming Data for Privacy Protection”, ICDE 2008, 978-1-4244-1837-4/08/$25.00 (© 2008 IEEE
8. Bin Zhou, Yi Han, Jian Pei, Bin Jiang, Yufei Tao and Yan Jia,, “Continuous Privacy Preserving Publishing of Data Streams”, Copyright 2009 ACM 978-1-60558-422-5/09/0003
9. Thanaa M. Ghanem, Ahmed K. Elmagarmid, Per-Ake Larson and Walid G. Aref, “Supporting Views in Data Stream Management Systems”, ACM Transactions on Database Systems, Vol. 35, No. 1, Article 1, Publication date: February 2010.
10. Jianneng Cao, Barbara Carminati, Elena Ferrari and Kian-Lee Tan, “CASTLE: Continuously
11. Anonymizing Data Streams”, IEEE Transactions on Dependable and Secure Computing, Vol. 8, No. 3, May/June 2011.
12. Barbara Carminati, Elena Ferrari, “A Framework to Enforce Access Control over Data Streams”. ACM Transactions on Computational Logic, Vol. V, No. N, November 2008.
13. Yongluan Zhou, Lidan Shou, and Xuan Shang, “Dissemination of Anonymized Streaming Data”, 2015 ACM 978-1-4503-3286-6/15/06.
14. David F. Ferraiolo, Ravi Sandhu, Serban Gavrila, D. Richard Kuhn And Ramaswamy Chandramouli, “Proposed NIST Standard for Role-Based Access Control”, ACM Transactions on Information and System Security, Vol. 4, No. 3, August 2001, Pages 224–274.
15. Shariq Rizvi, Alberto Mendelzon, S. Sudarshan and Prasan Roy, “Extending Query Rewriting Techniques for Fine Grained Access Control” *SIGMOD 2004* June 1318, 2004 Copyright 2004 ACM 1581138598/ 04/06
16. Anwar Dafa-Alla, Keun Ho Ryu, “PRBAC: An Extended Role Based Access Control for Privacy preserving Data mining”, Proceedings of the Fourth Annual ACIS International Conference on Computer and Information Science (ICIS’05) 0-7695-2296-3/05 © 2005 IEEE
17. Surajit Chaudhuri, Raghav Kaushik, and Ravi Ramamurthy, “Database Access Control & Privacy: Is There a Common Ground?” 5th Biennial Conference on Innovative Data Systems Research (CIDR ’11) January 9-12, 2011, Asilomar, California, USA.
18. Phuwanai Thummavet and Sangsuree Vasupongayya, “Privacy-preserving emergency access control for personal health records”, Maejo Int. J. Sci. Technol. **2015**, 9(01), 108-120; doi: 10.14456/mijst.2015.7.
19. Zahid Pervaiz, Walid G. Aref, Arif Ghafoor and Nagabhushana Prabhu, “Accuracy-Constrained Privacy-Preserving Access Control Mechanism for Relational Data”, IEEE Transactions On Knowledge And Data Engineering, Vol. 26, No. 4, April 2014.
20. Zahid Pervaiz, Arif Ghafoor, and Walid G. Aref, “Precision-Bounded Access Control Using Sliding-Window Query Views for Privacy-Preserving Data Streams”, IEEE Transactions On Knowledge And Data Engineering, Vol. 27, No. 7, July 2015. https://archive.ics.uci.edu/ml/datasets.html.