# Target-Based Privacy Preserving Association Rule Mining

Madhu Ahluwalia, Aryya Gangopadhyay[1], Zhiyuan Chen[1], Yelena Yesha
University of Maryland Baltimore County (UMBC)
1000 Hilltop Circle, Baltimore, MD 21250, USA
001.410.455.2620
{madhu.is, gangopad, zhchen}@umbc.edu, yeyesha@csee.umbc.edu

## ABSTRACT
Association rule mining is an important data mining task applicable across many commercial and scientific domains. There are instances when association analysis must be conducted by a third party over data located at a central point, but updated from several source locations. The source locations may not allow tracking changes. The target location must then take charge of the changed data detection and privatization process. We propose a solution to conduct privacy preserving association rule mining on such data. An evaluation of our approach shows that compared to existing approaches, it renders higher privacy, preserves 90% - 100% of the rules and is efficient for 10% database changes.

## Categories and Subject Descriptors
D.3.3 [**Data Mining**]: Data Mining

## General Terms
Experimentation, Algorithms

## Keywords
Privacy Preserving Data Mining, Discrete Wavelet Transform

## 1. INTRODUCTION
Association rule mining has been studied extensively in the context of preserving privacy of individual transactions. While many techniques have been presented for data that is owned by multiple parties and distributed over multiple sites, techniques that target data in the custody of a single owner and gathered at a single central site are not uncommon. We focus on the latter collection strategy assuming that the data owner's operational data is scattered over a few source sites, but collected at a single target site and that he himself lacks the capability to mine his collection of data. This assumption is often true in supply chain management environment where there are a large number of small retailers.

Against a backdrop of supply chain management, methods have been devised to protect the privacy of data when it is exchanged with a third party for the purpose of mining. One such method is proposed for association rules in [3], when the body of data in the possession of an owner undergoes no change at all. The essence of this method lies in its use of Discrete Wavelet Transform (DWT) on groups of similar numeric values so that sensitive values are masked and association rules generated on replicated wavelet coefficients (the transformed dataset) remain sufficiently close to the association rules on the original dataset.

Realistically, however, transactional source data that may be distributed over multiple sites is seldom static. The site that collects the data (e.g. a target database) must therefore be updated periodically in order to reflect changes occurring in source systems. An approach to synchronize a source location with a target location is described in [4]. It relies on comparing hash values for corresponding partitions of the source and the target databases so that if these values differ, the tuples in the target partition can be dropped and reloaded from the source. A table of partition information guides this process.

We borrow ideas from [4] and extend the approach presented in [3] to mine association rules on evolving databases. We assume that relatively few changes occur in the source databases and that they allow only read access to data. Further we assume that the schema is identical in all source databases. Finally, we assume that both the source and target databases are accessible for queries throughout the period in which sanitized data is updated.

Since data is exchanged between the data owner (the target database) and the data miner (an adversary), the former must find ways to maintain privatized data. Ideally, the solution is efficient, minimizes computation, storage and network resources consumed during privatization, and interoperates between diverse database management systems (DBMSs). A brute-force solution has the target database retrieve and transform all records in each table from each source database.

We propose a method that allows privacy preserving association rule mining on incrementally updated wavelet transformed (or privatized) data using target-only resources. We assume that after the very first synchronization, the target database secures a table of full partition information (summary of initial synchronization between the source and the target systems) and full transformation (replicated wavelet coefficients on first synchronization) on its own space. In the presence of changes to the source, the target database uses the content of the partition information table to retrieve complete raw tuples from the source database; we aim to decrease the number of retrieved tuples to decrease the number of updates performed on coefficients; For each table in the source database, the target database horizontally partitions the total order


*SAC'11*, March 21-25, 2011, TaiChung, Taiwan.


[1] This work was supported in part by the National Science Foundation under Grant Numbers IIS-0713345.

of tuples based on the partition start and end markers (row IDs of previous synchronization) of the partition information table, summarizes the tuples in each partition with a hash of all data in the partition, and matches this hash summary with the hash values in the table of partition definition. If the summaries match then we assume that the source table has not been subject to any change and the wavelet transforms do not require incremental update. If the summaries do not match then the target database retrieves all tuples in the partition's range from the source database, sorts each continuous-valued attribute in ascending order, applies Discrete Wavelet Transform over it, and stores and duplicates the approximation coefficient; thus providing for 2-anonymity in the attribute's transformed domain. The partition range that corresponds to the unmatched hash summary is then deleted in the previously transformed domain and the new coefficients are inserted in its place.

Section 2 refines the problem statement and describes related solutions. Section 3 provides the details of our algorithm and discusses some limitations. Section 4 presents our empirical tests of our algorithm before concluding in Section 5.

# 2. PROBLEM DEFINITION AND RELATED WORK

We consider the problem of mining quantitative association rules in a dynamic environment where the onus of tracking changes and incrementally updating privatized data is upon the target database because security concerns prevent source databases to allow access to their transaction logs or to alter their database schema or to add triggers. The source database simply responds to queries from the target database, much like any other client application that is accessing the source database.

The target database need not be empty when privatization happens. It simply seeks to update its privatized data. It does not synchronize the data and it does not strive for 100% accuracy in privatized data. Association rules generated on the incrementally updated transformed data must match the rules generated on full transformation of the synchronized data. Updated privatized data should not reveal updated original (or any other) data.

## 2.1 Related Work

Privacy preserving association rule mining using target-only resources has not been researched in the past. Nor does the literature contain work on mining association rules from incrementally updated perturbed data. However, a rich body of work exists on privacy preserving association rule mining for categorical data [7], Boolean data [6], in a distributed mining environment [10], in an environment where data is centralized [1], and where rules instead of data values are hidden [14].

There is little past work in capturing wavelet coefficients in a dynamic environment. By defining key relational set operators, Chakrabarti et al. [5] demonstrate how wavelet decomposition can be used in query processing. However, incremental maintenance of coefficients is not explored. In [9], the authors apply the properties of wavelets only to append new coefficients to existing coefficients. In contrast, our solution applies to both inserts as well as deletes. We view updates as deletes followed by inserts. A simple incremental modification of wavelet coefficients is proposed in [16], however not in the context of privacy preserving association rules.

# 3. Preliminaries

## 3.1 Association Rule Mining

We use Apriori, the classic algorithm for learning association rules. The algorithm first finds all the sets of items that appear frequently enough to be considered significant and it then derives from them the association rules that are strong enough to be considered interesting. The user determines the level of interestingness by specifying the minimum support threshold and minimum confidence threshold. Formal definitions of these metrics for an association rule of the form $X \rightarrow Y$ are $\frac{|X \cup Y|}{N}$ and $\frac{|X \cup Y|}{|X|}$ respectively. $X$ and $Y$ are disjoint itemsets and $N$ is the number of transactions in a dataset. Thus, while the support is a simple probability or a measure of the frequency of a rule, the confidence is a conditional probability or a measure of the strength of the relation between sets of items.

## 3.2 Discrete Wavelet Transforms

Discrete wavelet transform is used to analyze data by decomposing it into different frequency components and then studying each component with a resolution matched to its scale. We use the Haar wavelet due to its simplicity. For an input represented by a list of $2^n$ numbers, the Haar wavelet transform simply pairs up input values, storing the difference and passing the sum. The difference is called the detail coefficients and the sum is called the approximation coefficients[2]. This process can be repeated recursively over the approximation coefficients in the previous level. In our experiments, we apply one level of Haar wavelet transform.

We use DWT in our algorithm to preserve patterns and privacy of the original data. Our motivation of using DWT is based on several properties of DWT.

1. DWT allows a multi-resolution representation of discrete data. Important patterns in the high resolution data are usually preserved in the low resolution data (i.e., the approximation coefficients).
2. We can prune detail coefficients to provide privacy protection. Without the detail coefficients, the exact values of the original data cannot be reconstructed from the transformed data because there would be infinitely many solutions towards solving for the original data from the approximation coefficients.
3. DWT can be done very efficiently (in $O(n)$ time for n input data points).

### *3.2.1 Challenge of Using DWT in Our Context*

The main challenge of using DWT in the context of privacy preserving association rule mining is the order dependency of wavelet transforms. In other words, completely different sets of coefficients are generated if the dataset is ordered differently. We have devised a method that optimally groups similar data values together so that association rules can be preserved. The key idea is to sort the data on each attribute so that within-group data homogeneity is maximized.

[2] We divide the sum and difference by two in this paper.

# 4. METHOD

We have devised a method to mine association rules on data that is subject to change. This data resides in several source systems that may be limited by space, time and security constraints. A target system that is designated as a single point of data collection must then rely on its own resources to detect the change and incrementally mask the data for mining purposes. We show that association rules generated on this data are meaningful. We call this approach *Wavelet Coefficient Maintenance for privacy preserving association rules*.

We describe our algorithm in Section 4.1 and outline factors that affect its performance in Section 4.2. Section 4.3 presents limitations of our algorithm.

## 4.1 Algorithm

The proposed algorithm differs from existing approaches in that it accommodates the key data modification operations on the original data and preserves data privacy as well as rules.

In our coefficient maintenance algorithm, the target database poses queries to the source database to retrieve information and tuples. For maximum inter-operability, we restrict our queries to 1992 SQL standard [15], which appears to be the latest common SQL subset among most commercial vendors' DBMSs. The inter-operability is achieved at the cost of avoiding some SQL features that would provide for a more efficient query or partitioning on certain source DBMSs.

We describe the wavelet coefficient maintenance (WCM) algorithm for privacy preserving association rule mining in the context of a single database table. Multiple tables can be processed using one instantiation of the algorithm for each table, whether executed sequentially or concurrently. We assume that the source and the target databases are already synchronized and the target holds on its space a table of partition information of the current synchronization. The partition information refers to partition start and end markers (row IDs) and hashes of the current synchronization. Before releasing the present synchronization to a miner, the target perturbs it and stores a copy of this perturbation on its space. We now seek to directly update this perturbation in the presence of updates to the source.

The essence of the algorithm, shown in Figure 1, is to partition the source database table into variable-sized partitions using the partition start and end markers of the partition definition table, and then to compute the hash value of all tuples in each source partition. If the hash of the source partition differs from its counterpart in the partition definition, retrieve the partition from the source, sort it on each dimension in ascending order and apply the Haar wavelet transform on the sorted dimension. Retain only the approximation coefficient which is an average of two closest values. Since there are $n$ rows but only $n/2$ coefficients in each partition dimension, copy each coefficient to two consecutive rows. Update the previous transformation with new coefficients. Finally generate association rules on updated transformation.

Let $t_1 \ldots t_n$ be the tuples in the target database's synchronized table and $s_1 \ldots s_n$ be the tuples in the source database's table totally ordered by their primary keys. Let hash() be a hash function common to both databases, such as SHA-1. Figure 1 then describes the algorithm as executed at the target database.

**Algorithm: Wavelet Coefficient Maintenance**

**Input: Source Database**: Dataset S
**Target Database:** Transformed dataset T, number of tuples in a partition *r*, table of partition information containing total number of partitions *p*, partition ranges $t_{r\,(i1-1)+1} \ldots t_{ri1}, t_{r\,(i2-1)+1} \ldots t_{ri2}, \ldots$ and their corresponding hashes $h_{t1}, h_{t2}, \ldots$

**Output: Target Database:** updated transformed dataset T'

1. **for** i = 1 to *p* **do**
   1.1 Create an SQL query Q to extract all tuples $\geq t_{r(i-1)+1}$ and $< t_{ri+1}$ on S, catenate tuples into a local buffer B, and return hash(B)
   1.2 Send Q to S and retrieve the query result in $h_s$
   1.3 **if** ($h_t \neq h_s$) **then**
      1.3.1 Create an SQL query Q' to extract and return all tuples $\geq t_{r(i-1)+1}$ and $< t_{ri+1}$ on S
      1.3.2 Send Q' to S and retrieve the resulting tuples
      1.3.3 **for** each column $C_j \in s_{r\,(i-1)+1} \ldots s_{ri}$ **do**
         1.3.3.1 Sort $s_{r\,(i-1)+1} \ldots s_{ri}$ by column $C_j$
         1.3.3.2 Perform DWT, store and duplicate only approximation coefficients
         1.3.3.3 **if** r is odd in $s_{r\,(i-1)+1} \ldots s_{ri}$, rotate or eliminate last tuple
      1.3.4 **end**
      1.3.5 Delete tuples $t_{r\,(i-1)+1} \ldots t_{ri}$ from T
      1.3.6 Insert the tuples with updated coefficients from step 1.3.3 into T
   1.4 **end**
2. **end**
3. For all tuples from S smaller than $t_1$ and all tuples from S larger than $t_n$, apply steps 1.3.3 to 1.3.6
4. T transformed to T' and passed to miner to generate rules.

**Figure 1. Wavelet Coefficient Maintenance algorithm**

Depending on the implementation, the core of the WCM algorithm can be divided into five separate stages with each stage of the first four stages within a loop that iterates over all partitions: computing the hashes in the source database (steps 1.1 and 1.2), retrieving the updated records when hashes fail to match (steps 1.3.1 and 1.3.2), computing wavelet coefficients (step 1.3.3), updating the previously transformed dataset (steps 1.3.5 and 1.3.6), and generating association rules on the updated transformation (step 4). Our implementation computes and updates coefficients in the target immediately as differences are detected between source and target partitions.

Figure 2 illustrates the algorithm shown in Figure 1. We assume that the source and the target were once synchronized. Let the partition size be four rows per partition. The target database then stores on its space, tables of partition definition (top middle), and transformation (top right) of the initial synchronization (top left). The source now undergoes some change (bottom left). Hash values identify that partition 1 has changed, hence it is sorted on each attribute and a mean is computed over two consecutive values and replicated. The transformed partition (bottom right) is then plugged in the previous transformation (top right).

Source Dataset S

| RowID | Quantity | Price | Discount |
|---|---|---|---|
| 1 | 25.00 | 125.99 | 0.16 |
| 2 | 19.00 | 76.95 | 0.12 |
| 3 | 8.00 | 49.99 | 0.00 |
| 4 | 27.00 | 119.49 | 0.17 |
| 5 | 15.00 | 51.99 | 0.15 |
| 6 | 6.00 | 32.45 | 0.05 |
| 7 | 47.00 | 150.05 | 0.21 |
| 8 | 18.00 | 64.25 | 0.13 |
| 9 | 35.00 | 105.87 | 0.30 |
| 10 | 5.00 | 15.25 | 0.10 |

Partition Information of S

| PartID | RowID | Hash |
|---|---|---|
| 1 | 1 | H1H1 |
| 2 | 5 | H2H2 |
| 3 | 9 | H3H3 |

Transformed Dataset T

| RowID | Quantity | Price | Discount |
|---|---|---|---|
| 3 | 11.5 | 50.99 | 0.03 |
| 6 | 5.5 | 23.85 | 0.03 |
| 10 | 5.5 | 23.85 | 0.11 |
| 2 | 18.5 | 70.6 | 0.11 |
| 8 | 18.5 | 70.6 | 0.14 |
| 5 | 11.5 | 50.99 | 0.14 |
| 1 | 26 | 138.02 | 0.17 |
| 4 | 26 | 112.68 | 0.17 |
| 7 | 41 | 138.02 | 0.26 |
| 9 | 41 | 112.68 | 0.26 |

Updated Rows in S

| | | | |
|---|---|---|---|
| 1 | 29.00 | 130.75 | 0.17 |
| 2 | 20.00 | 86.95 | 0.12 |
| 4 | 28.00 | 123.49 | 0.19 |

Transformed Partition in T

| | | | |
|---|---|---|---|
| 3 | 14.00 | 68.47 | 0.06 |
| 2 | 14.00 | 68.47 | 0.06 |
| 1 | 28.50 | 127.12 | 0.18 |
| 4 | 28.50 | 127.12 | 0.18 |

**Figure 2. Example of executing the WCM algorithm.**

## 4.2 Performance Factors

The performance of WCM will depend on factors related with both synchronization and privatization. Synchronization factors favor fewer changes to numerous changes in the source [4]. Privatization factors favor implementing the DWT-based method without an intermediate results matrix [3].

## 4.3 Limitations

Our approach works only for numerical data types. This is because wavelet transforms cannot be performed on categorical and date data types. Boolean data types create a disclosure risk because input values of correlated items are restricted to 0 and 1. Since real-world datasets are of a mixed type, it may be feasible to use existing approaches on privacy preserving mining of association rules for categorical data in conjunction with our proposed approach to mitigate this limitation.

The incremental transformation is unlikely to be 100% accurate. This is the effect of rotating or eliminating the last row in transformed partitions that consist of an odd number of rows (see step 1.3.3.3 in figure 1). As a result there is bound to be some discrepancy between the dataset updated by incrementally capturing coefficients and the dataset that undergoes transformation from scratch. Another factor that affects accuracy is the sort order of attribute values in the changed partition. Sorting values within a partition will produce a different sort order than if they were sorted by first merging with the entire data. One might critique that the incremental approach scores low on accuracy, however, considering that it is proposed for large databases in which few changes occur, it might be prudent to relax the requirement of 100% accuracy in coefficient generation. Moreover, the updated coefficients are input in association rule mining, which returns aggregated or summarized information. The support and confidence of rules is not expected to alter drastically as a result of rotating, discarding or poorly sorting some rows.

# 5. EXPERIMENTAL RESULTS

Our experiments indicate that our proposed approach not only performs incremental updates on the transformed data efficiently when few changes are made to the raw data, but it also keeps the association rules generated from the updated transformed data very similar to the association rules generated from the fully transformed data. In so doing, it also keeps the privacy level of the raw data sufficiently high. Both the source and target databases are operational, while updates take place on the transformed data. Our method causes no overhead for the source and does not alter it in any way.

Section 5.1 describes the setup. Section 5.2 presents the results for pattern preservation. Section 5.3 reports the degree of privacy offered by our approach and the results of comparisons with existing approaches. Section 5.4 reports the execution time.

## 5.1 Setup

The experiments were conducted using Microsoft SQL Server 2005 installed on two machines. The algorithm was implemented in JAVA version 1.6. Privacy levels were measured using Matlab R2007b. For association rule mining, we used Oracle Data Miner 10.2 with Oracle application server 10g running on back-end.

**Datasets:** The experiments were run over four real datasets. Datasets "Adult" and "Pima Indians" were obtained from UCI Machine Learning Repository [8]. Datasets "Custom Checking" and "Sales Fact" are from the "Bank" and "Grocery" databases in the Ralph Kimball collection of databases [11]. Table 1 lists the characteristics of these datasets. All experiments were performed using numeric attributes.

**Table 1. Properties of datasets**

| Database Name | No. of Records | No. of Attributes |
|---|---|---|
| Adult | 32561 | 14 |
| Custom Checking | 5814 | 12 |
| Pima Indians | 768 | 8 |
| Sales Fact | 11040 | 8 |

**Privacy-preserving algorithms**: We used three privacy-preserving algorithms: WCM (Haar wavelet), kd-tree [12], and random projection [13]. The accuracy of association rules was determined for data captured and perturbed with the WCM algorithm. The privacy levels were compared for data perturbed by all three methods.

**Data mining parameters:** A minimum support of 0.1 and 0.3 and a minimum confidence of 0.5 and 0.7 were used for all datasets.

**Rule quality metrics:** We compared rules obtained from full transformation with rules obtained from updated transformation using the metrics recall and precision. Recall is defined as $\sum \frac{(T_R \cap T'_R)}{T_R}$ and precision is defined as $\sum \frac{(T_R \cap T'_R)}{T'_R}$ where $T_R$ denotes rules obtained from full transformation and $T'_R$ denotes rules obtained from incrementally updated transformation.

**Bin sizes:** To avoid overfitting, association analysis allows discretizing continuous attributes. Discretization replaces each continuous attribute value with its corresponding discrete interval. If the number of intervals is too large, there may be too few records in each interval to satisfy the minimum support and minimum confidence levels. On the other hand, if the number of intervals is too small, then some intervals may aggregate records which would otherwise generate unique rules. The process of specifying the number of intervals is referred to as binning. We

use equiwidth binning with bin sizes 2, 5 and 8 to study the effect of bin size on rule preservation.

**Partition sizes:** We used a fixed partition size of 100 rows per partition in all datasets.

**Percentage of changes:** Partitions in the source table were randomly selected to change for each experiment. We experimented with 10% and 50% changes to partitions in all datasets. Changes were generated randomly between the minimum and maximum values of each attribute.

We measure the effectiveness of our proposed approach on three levels: pattern preservation, privacy preservation and execution time. The details on each are as follows.

## 5.2 Preserving Patterns

To measure the quality of rules obtained from our approach we strove to find how closely they matched the rules generated from a full transformation of the synchronized dataset. The extent to which rules from a full transformation are preserved in the updated transformation can be measured by recall and precision. Recall measures the extent of patterns preserved and precision measures the extent of non-extraneous patterns generated. High recall and precision indicate good pattern preservation. We calculated recall and precision on each dataset for three bin sizes with 10% and 50% changes to partitions at $sup_{min}$ = 10% and $conf_{min}$ = 50%. Table 2 shows recall and precision for a bin size of 2. Table 3 shows recall and precision for a bin size of 5 and Table 4 shows recall and precision for a bin size of 8. The results at $sup_{min}$ = 30% and $conf_{min}$ = 70% are similar and thus omitted.

**Table 2. Accuracy of rules obtained from 2-bin discretization**

| Database Name | 10% Change | | 50% Change | |
|---|---|---|---|---|
| | Recall (%) | Precision (%) | Recall (%) | Precision (%) |
| Adult | 100 | 100 | 100 | 100 |
| Custom Checking | 89.47 | 89.31 | 89.93 | 90.18 |
| Pima Indians | 98.18 | 97.68 | 94.38 | 97.77 |
| Sales Fact | 100 | 100 | 100 | 100 |

**Table 3. Accuracy of rules obtained from 5-bin discretization**

| Database Name | 10% Change | | 50% Change | |
|---|---|---|---|---|
| | Recall (%) | Precision (%) | Recall (%) | Precision (%) |
| Adult | 77 | 59 | 91.7 | 58 |
| Custom Checking | 99.75 | 99.6 | 98.75 | 99.16 |
| Pima Indians | 95.18 | 99.12 | 88.3 | 91 |
| Sales Fact | 92.5 | 100 | 100 | 100 |

**Table 4. Accuracy of rules obtained from 8-bin discretization**

| Database Name | 10% Change | | 50% Change | |
|---|---|---|---|---|
| | Recall (%) | Precision (%) | Recall (%) | Precision (%) |
| Adult | 95.6 | 95.6 | 100 | 37.5 |
| Custom Checking | 98.95 | 99.1 | 100 | 99.29 |
| Pima Indians | 97.4 | 98.69 | 100 | 92.85 |
| Sales Fact | 86 | 87 | 100 | 100 |

The figures indicate the strength of the proposed approach in preserving rules at varying thresholds. We find that in general the recall and precision are both very high signifying excellent pattern preservation regardless of the bin sizes, the percentage of partitions changed and minimum support and confidence thresholds. This is because both the full transformation and the updated transformation are discretized before mining. This assigns nearly the same interval to nearly the same number of data points, thus keeping the correlations between items intact. We also see some low figures on precision for the adult dataset. This is explained by the large size of the dataset and the bins. The effects of low precision can be mitigated by filtering extraneous rules.

## 5.3 Preserving Privacy

A widely accepted and used measure of privacy in the literature is based on confidence intervals [2]. It states that if an original attribute $x$ can be estimated with $c\%$ confidence to lie in the interval $[x_1, x_2]$, then privacy is measured by the difference between the transformed data and the original data, normalized by the range of the original data, as follows: $\frac{x_2 - x_1}{\max\{x\} - \min\{x\}}$, where $\max\{x\}$ is the maximum value of attribute $x$ and $\min\{x\}$, its minimum value. We used the confidence interval measure to compare the privacy achieved by our approach with privacies offered by kd-tree and random projection approaches. We used 95% confidence interval in our experiments. Figure 3a shows the levels of privacy attained for 10% changes to source database partitions. Figure 3b shows the levels of privacy attained for 50% changes to source database partitions.

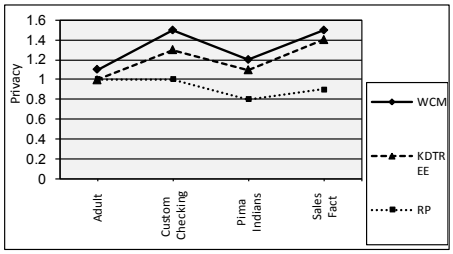


**Figure 3a. Privacy comparison with 10% database changes.**

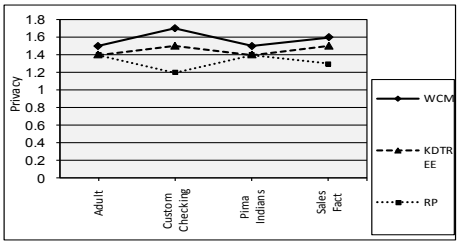


**Figure 3b. Privacy comparison with 50% database changes.**

The plots indicate that irrespective of the percentage of changes, our approach offers higher privacy than kd-tree and random projection. Intuitively, privacy should be higher when more data is changed. We see that in figure 3b.

## 5.4 Execution Time

There are two main overheads that interest us in the present scenario. The time it takes to detect and retrieve the changes in the source and the time it takes to transform and plug the transform in the previous transform. We measured these times for our approach when the changes to the number of partitions in the source database are 10% and 50%. We compared the total time to

the time it takes a brute-force approach to complete these operations. Figure 4 shows this comparison.

In the absence of our approach, one must resort to full replication with table-level locking and full transformation of the fully replicated data. There is no information on the last synchronization (partition start and end markers and hashes) that can be exploited for the current synchronization and transformation, so the fastest approach to transform the changed table would be to truncate the whole table and reload it from the source, then apply full transformation on the fully reloaded table, because an entire table cannot be transformed in memory. This would mean that the entire table is locked during the full replication and full transformation period, disabling queries to the target during that period.

Our experiments plotted in figure 4 show that our approach consistently outperforms full transformation with table-level locking when the source is subject to 10% changes, strengthening the efficacy of our approach for small changes to the source database. For 50% changes our approach shows time savings only for small datasets (datasets under 10,000 rows). We consider a 50% change to be a substantial change not ideal for our proposed approach.

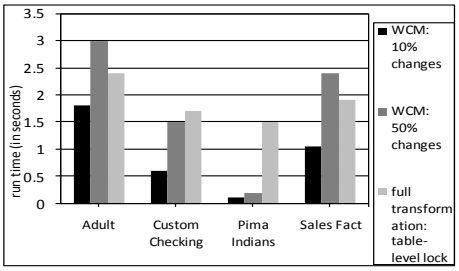


**Figure 4. Execution time with 10% and 50% database changes.**

## 6. CONCLUSION AND FUTURE WORK

Previous work on privacy preserving quantitative association rule mining has been set in an environment where the collection of data is not subject to change. Since data is seldom static, this approach has little practical significance. We propose a solution to mine association rules securely in a dynamic environment. Tests show that in most datasets 90% -100% of the rules can be preserved even when 50% of the source database undergoes a change. Comparisons indicate that the proposed WCM algorithm outperforms other approaches in preserving data privacy and for 10% of database changes in execution time.

Our future work will focus on maintaining mining results efficiently in such setting.

## 7. REFERENCES


[1] C. C. Aggarwal and P. S. Yu, "A Condensation Approach to Privacy Preserving Data Mining," *9th International Conference on Extending Database Technology*, pp. 183-199, 2004.

[2] R. Agrawal and R. Srikant, "Privacy preserving data mining," *2000 ACM SIGMOD Conference on Management of Data*, pp. 439-450, 2000.

[3] M. Ahluwalia, A. Gangopadhyay, and Z. Chen, "Preserving Privacy in Mining Quantitative Association Rules," *International Journal of Information Security and Privacy*, accepted 2010.

[4] M. Ahluwalia, R. Gupta, A. Gangopadhyay, Y. Yesha, and M. McAllister, "Target-Based Database Synchronization," presented at the 25th ACM Symposium on Applied Computing, Sierre, Switzerland, 2010.

[5] K. Chakrabarti, M. Garofalakis, R. Rastogi, and K. Shim, "Approximate query processing using wavelets " *The VLDB Journal* vol. 10 pp. 199-223 2001.

[6] Z.-Y. Chen and G.-H. Liu, "Quantitative Association Rules Mining Methods with Privacy-preserving," *Sixth International Conference on Parallel and Distributed Computing, Applications and Technologies, 2005. PDCAT 2005.*, pp. 910-912, 2005.

[7] A. Evfimevski, R. Srikant, R. Agrawal, and J. Gehrke, "Privacy preserving mining of association rules," *8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'02)*, pp. 217 - 228, July 2002.

[8] S. Hettich, C. Blake, and C. Merz, "UCI repository of machine learning databases, {http://www.ics.uci.edu/~mlearn/MLRepository.html}," 1998.

[9] M. Jahangiri, D. Sacharidis, and C. Shahabi, "SHIFT-SPLIT: I/O efficient maintenance of wavelet-transformed multidimensional data," in *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*. Baltimore, Maryland: ACM, 2005, pp. 275-286.

[10] M. Kantarcioglu, and C. Clifton, "Privacy-preserving distributed mining of association rules on horizontally partitioned data, " *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, pp. 1026-1037, 2004.

[11] R. Kimball, and M. Ross, *The Data warehouse Toolkit*. Wiley, 2002.

[12] X.-B. Li and S. Sarkar, "A Tree-Based Data Perturbation Approach for Privacy-Preserving Data Mining," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 18, pp. 1278-1283, 2006.

[13] K. Liu, H. Kargupta, and J. Ryan, "Random projection-based multiplicative data perturbation for privacy preserving distributed data mining," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 18, pp. 92-106, 2006.

[14] S. Oliveira and O. R. Zaiane, "Privacy Preserving Frequent Itemset Mining," presented at IEEE International Conference on Privacy, Security and Data Mining, Japan, 2002.

[15] A. Silberschatz, H. Korth, and S. Sudarshan, *Database System Concepts (5th Ed.)*: McGraw-Hill, 2006.

[16] Y.-B. Xiong, B. Liu, and Y.-F. Hu, "Approximate Query Processing Based on Wavelet Transform," *International Conference on Machine Learning and Cybernetics*, pp. 1337-1342, 2006