Privacy-Preserving Classification of Data Streams

Ching-Ming Chao[1], Po-Zung Chen[2] and Chu-Hao Sun[2]

[1]Department of Computer Science and Information Management, Soochow University,
Taipei, Taiwan 100, R.O.C.
[2]Department of Computer Science and Information Engineering, Tamkang University,
Tamsui, Taiwan 251, R.O.C.

Abstract

Data mining is the information technology that extracts valuable knowledge from large amounts of data. Due to the emergence of data streams as a new type of data, data streams mining has recently become a very important and popular research issue. There have been many studies proposing efficient mining algorithms for data streams. On the other hand, data mining can cause a great threat to data privacy. Privacy-preserving data mining hence has also been studied. In this paper, we propose a method for privacy-preserving classification of data streams, called the PCDS method, which extends the process of data streams classification to achieve privacy preservation.

The PCDS method is divided into two stages, which are data streams preprocessing and data streams mining, respectively. The stage of data streams preprocessing uses the data splitting and perturbation algorithm to perturb confidential data. Users can flexibly adjust the data attributes to be perturbed according to the security need. Therefore, threats and risks from releasing data can be effectively reduced. The stage of data streams mining uses the weighted average sliding window algorithm to mine perturbed data streams. When the classification error rate exceeds a predetermined threshold value, the classification model is reconstructed to maintain classification accuracy. Experimental results show that the PCDS method not only can preserve data privacy but also can mine data streams accurately.

***Key Words*:** Data Streams, Data Mining, Classification, Privacy Preservation, Incremental Mining

1. Introduction

Data mining is an information technology that extracts valuable knowledge from large amounts of data. Recently, data streams are emerging as a new type of data, which are different from traditional static data. The characteristics of data streams are as follows [1]: (1) Data has timing preference (2) Data distribution changes constantly with time (3) The amount of data is enormous (4) Data flows in and out with fast speed (5) Immediate response is required.

These characteristics create a great challenge to data mining. Traditional data mining algorithms are designed for static databases. If the data changes, it would be necessary to rescan the database, which leads to long computation time and inability to promptly respond to the user. Therefore, traditional algorithms are not suitable for data streams and data streams mining has recently become a very important and popular research issue.

Although data mining can discover valuable knowledge, it can also cause a great threat to data privacy. Clifton and Marks [2] are the first who pointed out the security and privacy problems of data mining. To preserve data privacy during data mining, the issue of privacy-preserving data mining has been widely studied and many techniques have been proposed. However, existing techniques for privacy-preserving data mining are

*Corresponding author. E-mail: chao@csim.scu.edu.tw designed for traditional static databases and are not suitable for data streams.

The privacy preservation issue of data streams mining is a very important issue. In this paper, we propose a method for privacy-preserving classification of data streams, called the PCDS method, which extends the process of data streams classification to achieve privacy preservation. The PCDS method is divided into two stages, which are data streams preprocessing and data streams mining, respectively. In the stage of data streams preprocessing, upon receiving data streams from sensor devices, the data streams preprocessing system uses the data splitting and perturbation algorithm to perturb confidential data. Users can flexibly adjust the data attributes to be perturbed according to the security need. Therefore, threats and risks from releasing data can be effectively reduced. In the stage of data streams mining, the online data mining system uses the weighted average sliding window algorithm to mine perturbed data streams. When the classification error rate exceeds a predetermined threshold value, the classification model is reconstructed to maintain classification accuracy. Experimental results show that the PCDS method not only can preserve data privacy but also can mine data streams accurately.

The remainder of this paper is organized as follows. Section 2 reviews related work. In Section 3 we present the PCDS method. In Section 4 we evaluate the performance of the PCDS method. Section 5 concludes this paper.

2. Related Work

2.1 Classification of Data Streams

According to the way training data are obtained, the construction of a classification model can be distinguished into non-incremental learning and incremental learning. In non-incremental learning, after all data are completely collected, some of the data are selected as the training data to construct a classification model. This way of learning has higher computation cost and is unable to satisfy user requirements that need immediate response. In incremental learning, in contrast, not all of the training data are completely collected at once. Data that have been collected are used to construct a classification model, and then newly collected data are used to modify the classification model. With incremental learning the classification model can fit in the newest situation [3].

In the past, most of the classification applications adopted non-incremental learning. However, for several new applications, such as e-mail classification, schedule planning, intrusion detection, sensor networks, etc., non-incremental learning is not appropriate due to the inability to obtain complete training data before constructing the classification model. If it is necessary to reconstruct the classification model whenever new data are obtained, the cost of model construction will increase tremendously. On the contrary, modifying the classification model to adapt to new data is a more efficient and feasible way.

There are three categories of incremental learning. The first category is learning without keeping instances [4]. Whenever new data are obtained, old data are abandoned. However, the classification model is not completely abandoned. Instead, new data are incorporated into the classification model. The disadvantage is that the classification model will forget some previously learned cases. Besides, the same training data set may produce different classification rules or decision trees because the order of obtaining data is different. The second category is learning with partial instance memory. Maloof and Michalski [5] proposed the AQ-PM learning method, which stores data located near the rule boundary. Upon arrival, new data are combined with stored data as training data to modify the classification model. The third category is learning with complete instances [6]. During the learning process, all stream data are preserved, and the data that are used to determine if the test attribute is still the best attribute are stored in each node. Upon arrival, new data are checked along with old data. If the test attribute is no longer the best attribute, some kind of modification mechanism will be activated to replace the test attribute. In addition, Street and Kim [7] developed a streaming ensemble algorithm for classification. First, the algorithm splits data into several fixsized continuous chunks. Then, it constructs a classification model for each individual chunk. Finally, an ensemble classification model is constructed by combining several individual classification models.

The above mentioned methods are mainly for reducing the learning cost. For large amounts of data streams, it is also necessary to take the leaning time into consideration. Domingos and Hulten [8] proposed the VFDT

(Very Fast Decision Tree Learner) algorithm to solve the problem of long learning time. The VFDT algorithm belongs to the third category of incremental learning and uses the statistical results of the Hoeffding bounds [9] to determine using fewer samples if the difference between the gain value of the best attribute and that of the second best test attribute is greater than a deviation value. When it is the case, it indicates that the best test attribute in the sample data can be used as the best test attribute of the whole data. Using this attribute as the test attribute in the root node, the remaining data are mapped to the leaf nodes according to the test in the root node and are used to select the test attributes in the leaf nodes. The main drawback of the VFDT algorithm is its inability to handle data distribution from different time. For many applications, new data are usually more important than old data. The VFDT algorithm does not consider the time of data, and hence cannot mine data from different time. Gama et al. [10] proposed the VFDTc algorithm, which improves the VFDT algorithm in two aspects: the ability to process continuous values in the leaf nodes and the usage of a more powerful classification technique. The VFDTc algorithm can preserve data for a long time and adjust the way data are stored in the database. However, it still has some drawbacks. For instance, in some applications users may only be interested in data that arrive in a certain period of time. Therefore, Hulten et al. [11] proposed the CVFDT algorithm, which not only extends the characteristics of the VFDT algorithm, but also improves the drawback of assuming data are stably distributed. The CVFDT algorithm attaches a sliding window, which contains a fixed amount of data and will remove old data as new data are added, to the training data set and constantly monitors the effect of the training data in the sliding window on classification accuracy of the current decision tree. As a result, data of any time within the sliding window can be mined, so as to satisfy various mining requirements of different time. The algorithm proposed in this paper is based on the CVFDT algorithm.

2.2 Privacy-Preserving Data Mining

Privacy-preserving data mining does not mean to restrict collection of data or application of information technology on data. Its primary objective is to achieve balance between privacy preservation and knowledge discovery. Therefore, the approaches should be designed not only to discover useful knowledge but also to preserve data privacy. Verykios et al. [12] classified privacy-preserving data mining techniques based on five dimensions, which are data distribution, data modification, data mining algorithms, data or rule hiding, and privacy preservation, respectively. We analyze the adaptability of various privacy-preserving data mining techniques to data streams below.

In the dimension of data distribution, some approaches have been proposed for centralized data and some for distributed data. Distributed data can be classified into horizontal distribution and vertical distribution. Horizontal distribution means that different records in a file may be scattered over several sites, while vertical distribution means that different attributes in a file may be scattered over several sites. Distributed data usually uses distributed data mining. In distributed data mining, data in different sites are mined separately to produce partial results, which are then integrated to produce the complete result. Du and Zhan [13] utilized the secure union, secure sum and secure scalar product to prevent the original data of each site from revealing during the mining process. At the end of the mining process, every site will obtain the final result of mining the whole data. The advantage of this approach is that each site cannot infer data of other sites because each site only holds part of the data. The disadvantage is that the approach requires multiple scans of the database and hence is not suitable for data streams, which flows in fast and requires immediate response.

In the dimension of data modification, the confidential values of a database to be released to the public are modified to preserve data privacy. Adopted approaches include perturbation, blocking, aggregation or merging, swapping, and sampling. Agrawal and Srikant [14] used the random data perturbation technique to protect customer data and then constructed the decision tree. The data receiver used the data distribution after perturbation to estimate the original data distribution, established a result approximate to that of the original data, and used this approximate result for data mining to obtain a classification model. However, the random data perturbation technique can only be applied to traditional databases. For data streams, because data are produced at different time, not only data distribution will change with time, but also the mining accuracy will decrease for modified data.

In the dimension of data mining algorithms, most of the previous research is focused on three kinds of data mining techniques: classification [15], clustering [16], and association analysis [17]. These studies designed various mining algorithms to preserve data privacy based on different mining requirements. Those data mining algorithms assume that large amounts of data are stored in the database to be mined and hence are not suitable for data streams that change frequently. In the dimension of data or rule hiding, some data or patterns are hided to preserve data privacy [18]. By reducing the amount of data revealed, the data mining site cannot easily infer confidential data from revealed data. However, this kind of method often overemphasizes data security and hence may sacrifice the accuracy of data mining.

In the dimension of privacy preservation, while considering the issue of data privacy, in order to make data with better quality and usability after modification, it is necessary to perform selective data modification, which can be achieved by means of privacy preservation techniques. Privacy preservation techniques can be classified into three categories, which are heuristic-based techniques, cryptography-based techniques, and reconstruction-based techniques. Techniques in the first category select appropriate data for adjustment after producing mining results to reduce impact on data usability to the lowest degree. In the second category of techniques, Kantarcioglou and Clifton [19] used the secure multiparty protocol to preserve data privacy. To exchange data, the sender and the receiver must use the same key and pass a confirmation procedure. Communication between the sender and the receiver is very complicated, which leads to higher communication costs. Therefore, this technique is suitable for data streams that require continuous transmission. Techniques in the last category use the random technique to change data distribution, and then reestablish the original data distribution from the changed data distribution.

From the review of previous research, we can see that existing techniques for privacy-preserving data mining are designed for static databases with an emphasis on data security. These existing techniques are not suitable for data streams.

3. The PCDS Method

3.1 The Overall Process

Figure 1 illustrates the overall process of the PCDS method for privacy-preserving classification of data streams. The process is divided into two stages, which are data streams preprocessing and data streams mining, respectively. The primary objective of the first stage, which is handled by the data streams preprocessing system (DSPS), is to perturb data streams to preserve data privacy. The primary objective of the second stage, which is handled by the online data mining system (ODMS), is to mine perturbed data streams to construct an accurate classification model.

Data streams continuously flow in DSPS and the arriving time of data is unpredictable. If DSPS processes data streams immediately upon arrival of the data, this will consume a lot of system resources. Therefore, DSPS adopts the batch processing mode to process incoming data streams. Not only system resources can be more effectively utilized, but also data mining can be more efficiently performed. Whenever accumulating a sufficient amount of data, DSPS uses the data splitting and perturbation algorithm to perturb confidential data as well as computes the error rate resulted from data perturbation. Then DSPS passes perturbed data and the error rate to ODMS.

ODMS uses the weighted average sliding window algorithm to mine perturbed data streams to construct a classification model. Because only partial data are available for data mining, ODMS utilizes the Hoeffding bounds sampling method to construct the classification model. In addition, ODMS adopts the sliding window mode to store and process received data streams. There are two reasons for adopting the sliding window model. First, the amount of data streams is enormous and hence

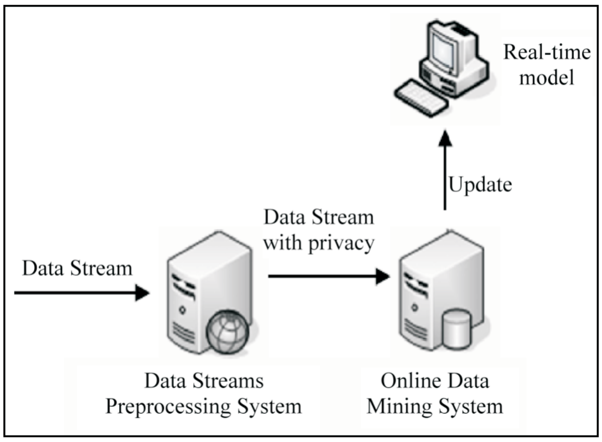

Figure 1. Overall process of the PCDS method.

it is impossible to store all data. Second, users are usually more interested in more recent data. When data distribution results in a significant change, ODMS reconstructs the classification model to keep it accurate

3.2 Data Streams Preprocessing

The primary objective of the stage of data streams preprocessing is to perturb data streams to preserve data privacy. Because data streams continuously flow in DSPS and the arriving time of data is unpredictable, DSPS is unable to collect the complete data and hence cannot use traditional perturbation techniques to perturb data streams. In addition, the data distribution of data streams can be different in different time. Using traditional perturbation techniques on data streams will increase the data error and hence will produce inaccurate mining results. As a result, whenever accumulating a sufficient amount of data, DSPS uses the data splitting and perturbation (DSP) algorithm to perturb confidential data. The DSP algorithm selects non-confidential attributes as the splitting attributes to partition the dataset. After the partition is completed, each value of each confidential attribute to be perturbed is replaced by the average value of those attribute values in its partition. When there are more non-confidential attributes used as the splitting attributes, the dataset will be partitioned into smaller subsets and the distribution of data in the same partition will be more similar. Therefore, compared to existing data perturbation techniques, the DSP algorithm has higher security and less data error. Finally, DSPS passes perturbed data to ODMS.

Figure 2 shows the steps of the DSP algorithm, which are described as follows. The initial step is inputting the original dataset *S* and prepares to construct a tree by splitting *S*. Non-confidential attributes in *S* will be used as the splitting attributes. Initially, the tree starts as a single node containing all records in *S*. The first step is to select a non-confidential attribute as the splitting attribute of the current node. We use *NA* be the set of non-confidential attributes. Second, compute the variance of each non-confidential attribute based on the records contained in the current node. Select the attribute, say j^*, which has the maximum variance as the splitting attribute. This step is to determine the splitting criterion and then partition the records contained in the current node into two disjoint subsets of records.

When the splitting criterion is determined by finding the median (or mid-range) of the splitting attribute, two child nodes are generated from the current node. Each child node contains a partition of the records j^* in the current node. This step is to complete the partition of *S*. Fourth, if *S* is completed partition, DSP repeat step 2 and 3 for each node generated in step 3 until a terminating condition is reached. This step is to perturb the confidential data in *S* and stops partitioning a node when the node contains less than a pre-specified number of records or no splitting attributes are available.

The fifth step is perturbed each confidential attribute values in each partition and replaces by their average value. For a leaf *t* with n_t records, let x_{t1}, …., x_{tn_t} be the values of the confidential attribute. Perturb the data by replacing these values with their average. Repeat for each leaf in the tree. Finally, return the perturbed dataset *S'* and pass it to ODMS.

We now use an example to demonstrate the DSP algorithm. Table 1 shows a sample dataset that has four at-

```
Algorithm: Data Splitting and Perturbation
Input: an original dataset $S$
Output: a perturbed dataset $S'$
Steps:
  1. Let $NA$ be the set of non-confidential attributes
     in $S$. The tree starts as a single node containing
     all records in $S$.
  2. Compute the variance of each attribute in $NA$.
     Let $j^*$ be the attribute with the maximum variance.
  3. Find the median (or mid-range) of attribute $j^*$.
     Partition the dataset in the current node into two
     subsets (child nodes) based on the median (or
     mid-range).
  4. Repeat steps 2 and 3 for each of the two child
     nodes. Stop partitioning a node when the node
     contains less than a pre-specified number of
     records or no splitting attributes are available.
  5. For each confidential attribute to be perturbed, do
     the following.
     For a leaf $t$ with $n_t$ records, let $x_{t1},....,x_{tn_t}$ be the
     values of the confidential attribute. Perturb the
     data by replacing these values with their average
     $\overline{x}_t = (1/n_t)\sum_{k=1}^{n_t} x_{tk}$ . Repeat for each leaf in the
     tree.
  6. Return the perturbed dataset $S'$.
```

Figure 2. DSP algorithm.

Table 1. Sample dataset

Record No.	age	education-level	salary (before perturbation)	salary (after perturbation)	has-computer
1	23	15	53	57.5	Yes
2	31	14	55	52.0	Yes
3	33	18	62	57.5	Yes
4	36	11	49	52.0	No
5	42	15	63	62.0	Yes
6	48	18	70	71.5	Yes
7	50	14	57	62.0	No
8	52	18	73	71.5	Yes
9	55	15	66	62.0	No

tributes and nine records. Among the four attributes, age, education-level, and has-computer are non-confidential attributes and salary is the only confidential attribute. Initially, the tree starts as a single node containing nine records. Because the age attribute has the maximum variance among the three non-confidential attributes, it is selected as the splitting attribute. Because the median of the age attribute is 39, the splitting criterion is age < 39. Partition this dataset into two subsets by generating two child nodes. Repeat the process for each of the two child nodes. The education-level attribute is selected as the splitting attribute.

Figure 3 illustrates the data partitioning process. After the partitioning process is completed, the values of the confidential attribute salary in each leaf node are replaced with their average value. For example, the leftmost leaf node, which is generated by the condition (age < 39 and education-level < 14.5), contains two records (#2 and #4) whose salary values are 55 and 49, respectively. These two values are replaced with their average value 52.

3.3 Data Streams Mining

The primary objective of the stage of data streams mining is to mine perturbed data streams to construct an accurate classification model. ODMS uses the weighted average sliding window (WASW) algorithm, which is an extension of the VFDT algorithm, to mine perturbed data streams. Figure 4 shows the steps of the WASW algorithm. Input to the algorithm is a sequence of perturbed datasets. The algorithm adopts the sliding window mode to store received datasets and assigns different weights to different datasets according to the order of arrival. Because the value of newer data is higher than that of older data, assigning larger weights to newer data can better reflect current data distribution. Because only partial data are available for data mining, the algorithm utilizes the Hoeffding bounds sampling method to efficiently construct the classification model. Each received dataset is input to the classification model to calculate its classification error rate. A threshold value of the error rate is predetermined. The algorithm calculates the weighted average error rate of the datasets in the sliding window. When the weighted average error rate exceeds the predetermined threshold value, the algorithm will reconstruct the classification model to keep the classification model accurate.

We now use an example to demonstrate the WASW algorithm. Figure 5 shows a sliding window W of size five. Each dataset in the data stream is assigned a different time weight; e.g., 0.01, 0.02, etc. The algorithm uses the first received dataset to construct a classification model and then calculates the classification error rate of the first dataset, which is 5%. The error rates of the following four datasets are calculated as 4%, 2%, 4%, and 1%, respectively. Because W is now full, the algorithm calculates the weighted average error rate $\overline{\mu}$ of the five datasets in W. Because $\overline{\mu}$ does not exceed the predeter-

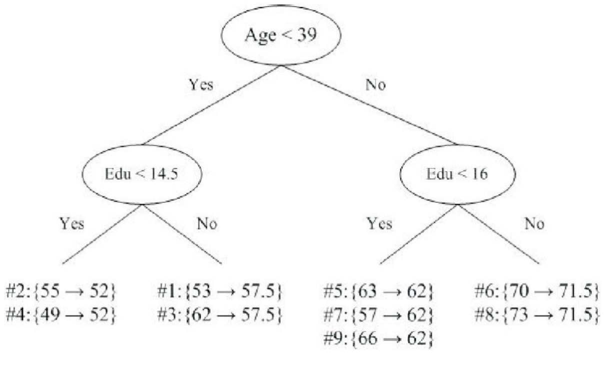

Figure 3. Partitioned and perturbed data.

```
Algorithm: Weighted Average Sliding Window
Input: a sequence of perturbed datasets
Output: a classification model DT
Steps:
1. Let $W$ be a sliding window of size $w$. Let $Hb()$ be a split evaluation function. Let $\rho$ be a threshold of the error rate.
2. Use $Hb()$ to construct a classification model $DT$.
3. Store next dataset in $W$ and calculate the error rate $\mu$ of applying $DT$ on this dataset.
4. Repeat step 3 until $W$ is full.
5. Calculate the weighted average error rate $\mu$ in $W$
       $\bar{\mu} = (1/w)\sum_{i=1}^{w} \mu_i \delta_i$
   where $\delta_i$ 's are time weights of datasets.
6. If $\bar{\mu} > \rho$, reconstruct a new classification model $DT$.
7. Remove the oldest dataset from $W$. Store next dataset in $W$ and calculate the error rate $\mu$.
8. Return the classification model $DT$ and wait the next stream data.
```

Figure 4. WASW algorithm.

mined threshold value, the algorithm removes the first dataset from *W* and stores the sixth dataset in *W*. Recalculate the weighted average error rate $\bar{\mu}$. Because $\bar{\mu}$ now exceeds the threshold value, the algorithm uses the sixth dataset to reconstruct a classification model to reflect current data distribution.

4. Performance Evaluation

In this section, we evaluate by experiment the performance of the PCDS method. For data streams preprocessing, we compare the security and data error between the DSP algorithm and four existing data perturbation algorithms SAN [20], MN [21], UMA [22], and MMA [23]. For data streams mining, we compare the accuracy between the WASW algorithm and the VFDT algorithm. Experimental data consist of five datasets, four of which are real world datasets and one of which is a virtual dataset generated by the synthetic data generator developed by the IBM Almaden Research Center.

4.1 Security Measurement

We use the average squared distance (ASD) and the distance-based record linkage (DBRL) between the original data and the perturbed data to measure the security of the DSP algorithm.

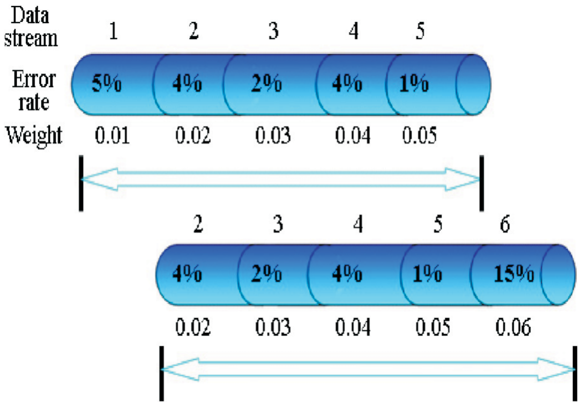

Figure 5. Weighted average sliding window.

$$ASD = \frac{1}{N}\sum_{i=1}^{N}(y_i - x_i)^2 \tag{1}$$

$$DBRL = \sum_{i=1}^{n}\left(\frac{x_i - \bar{x}}{\sigma(x_i)} - \frac{y_i - \bar{y}}{\sigma(y_i)}\right)^2 \tag{2}$$

x_i's are the original confidential values; y_i's are the perturbed values; N is the number of data records; $\bar{x}$ is the mean of x_i's; $\bar{y}$ is the mean of y_i's; $\sigma(x_i)$ is the standard deviation of x_i's; $\sigma(y_i)$ is the standard deviation of y_i's. ASD uses the space distance formula to measure the difference between the original data and the perturbed data. In addition to calculating the distance between two collections of data, DBRL also takes the standard deviation into account. Therefore, it can measure the variance level between the original data and the perturbed data.

Figure 6 shows the comparison of ASD measurement among the DSP algorithm and four other data perturbation algorithms using five different datasets. In all five datasets, the DSP algorithm has higher ASD values than other algorithms; therefore, it has higher security. Notice that the ASD values in the fifth dataset are lower than their corresponding ASD values in other four datasets. It is because there are less numeric attributes that can be used to perturb data in the fifth dataset. From this we can see that, in the process of perturbation, the number of numeric attributes is an important criterion to determine the risk level of data leakage. When there are more numeric attributes, data will be perturbed more seriously; therefore, the risk of data leakage will be lower.

Figure 7 shows the comparison of DBRL measurement among the DSP algorithm and four other data perturbation algorithms using five different datasets. In all five datasets, the DSP algorithm has lower DBRL values than other algorithms, which means that the correlation between the original data and the perturbed data is lower for the DSP algorithm. Therefore, it has a lower chance to infer the original data from the data perturbed by the DSP algorithm and so the DSP algorithm has higher security.

4.2 Data error Measurement

In addition to security, we also consider the data error of the mining results between the perturbed data and the original data. We use the bias in mean (BIM) and the bias in standard deviation (BISD) between the original data and the perturbed data to measure the data error of the DSP algorithm.

$$BIM = \left(\frac{\overline{Y} - \overline{X}}{\overline{X}}\right) \tag{3}$$

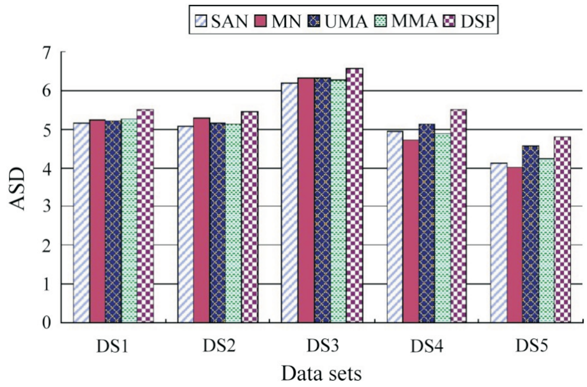

Figure 6. Comparison of ASD measure.

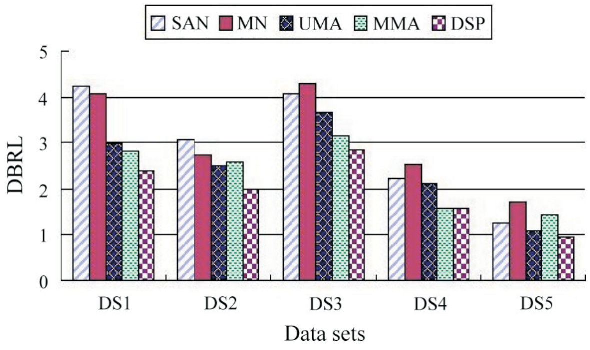

Figure 7. Comparison of DBRL measure.

$$BISD = \left(\frac{S_Y - S_X}{S_X}\right) \tag{4}$$

$\overline{X}$ is the mean of the original data; $\overline{Y}$ is the mean of the perturbed data; S_X is the standard deviation of the original data; S_Y is the standard deviation of the perturbed data. BIM calculates the difference of mean between the original data and the perturbed data to measure the data error. BISD calculates the difference of variance to measure the data error. Figure 8 and Figure 9 show the comparison of BIM measurement and BISD measurement, respectively. The DSP algorithm has lower BIM and BISD values than other algorithms in most cases. Therefore, the DSP algorithm has less data error.

4.3 Accuracy Measurement

We compare the error rate of mining perturbed data between the WASW algorithm and the VFDT algorithm. The threshold value of the error rate in the WASW algorithm is set to 15%. Figure10 shows experimental results

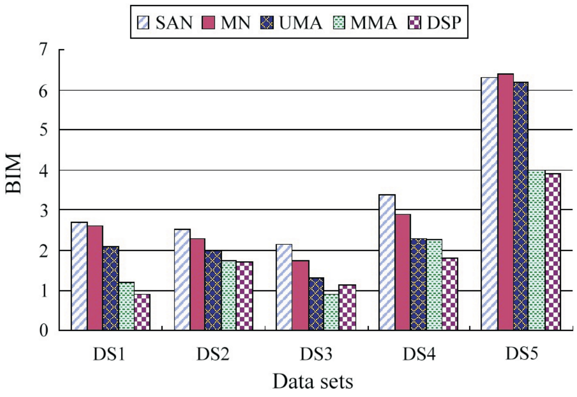

Figure 8. Comparison of BIM measure.

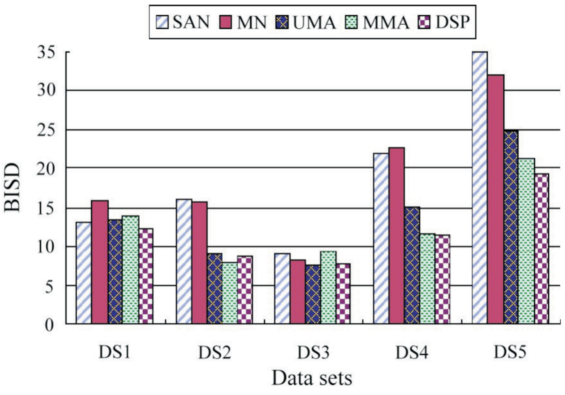

Figure 9. Comparison of BISD measure.

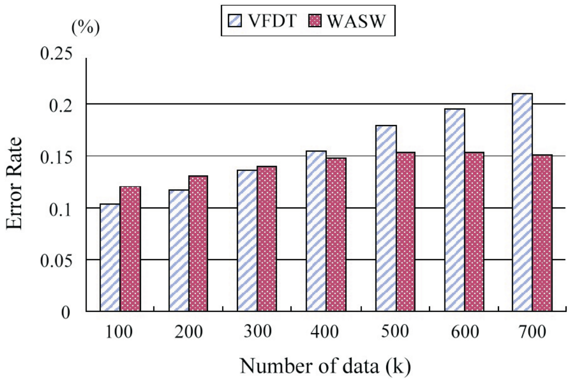

Figure 10. Comparison of the error rate.

on various data volumes. The initial error rate of the VFDT algorithm is 10%. Along with continuous arrival of the data stream, the error rate will increase constantly. On the other hand, although the initial error rate is 12%, the WASW algorithm will reconstruct the classification model to reduce the error rate when the error rate exceeds the predetermined threshold value. Therefore, the WASW algorithm can adjust to current data distribution to maintain the accuracy of the classification model.

5. Conclusion

In this paper we proposed the PCDS method for privacy-preserving classification of data streams, which consists of two stages: date streams preprocessing and data streams mining. In the stage of data streams preprocessing, we proposed the DSP algorithm to perturb data streams. Experimental results of security measurement showed that the DSP algorithm has higher ASD values and lower DBRL values than other data perturbation algorithms. Therefore, the DSP algorithm has higher security. Experimental results of data error measurement showed that the DSP algorithm has lower BIM and BISD values than other algorithms in most cases. Therefore, the DSP algorithm has less data error. In the stage of data streams mining, we proposed the WASW algorithm to mine perturbed data streams. Experiment results of accuracy measurement showed that the error rate of the VFDT algorithm increases constantly along with continuous arrival of the data stream but the error rate of the WASW algorithm is kept under the predetermined threshold value. Therefore, the WASW algorithm has higher accuracy. In conclusion, the PCDS method not only can preserve data privacy but also can mine data streams accurately.

Acknowledgment

The authors would like to express their appreciation for the financial support from the National Science Council of Republic of China under Project No. NSC 96-2221-E-031-001-MY2.

References

[1] Golab, L. and Ozsu, M., "Issues in Data Stream Management," *ACM SIGMOD Record,* Vol. 32, pp. 5–14 (2003).

[2] Clifton, C. and Marks, D., "Security and Privacy Implications of Data Mining," *Proceedings of ACM SIGMOD Workshop on Data Mining and Knowledge Discovery,* pp. 15–19 (1996).

[3] Utgoff, P. E., "Incremental Induction of Decision Trees," *Machine Learning,* Vol. 4, pp. 161–186 (1989).

[4] Schlimmer, J. C. and Fisher, D. H., "A Case Study of Incremental Concept Induction," *Proceedings of the 5th International Conference on Artificial Intelligence,* pp. 496–501 (1986).

[5] Maloof, M. A. and Michalski, R. S., "Incremental Learning with Partial Instance Memory," *Foundations of Intelligent Systems,* Vol. 2366, pp. 16–27 (2002).

[6] Jin, R. and Agrawa, G., "Efficient Decision Tree Construction on Streaming Data," *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* pp. 571–576 (2003).

[7] Street, W. and Kim, Y., "A Streaming Ensemble Algorithm (SEA) for Large-Scale Classification," *Proceedings of the 7th International Conference on Knowledge Discovery and Data Mining,* pp. 377–382 (2001).

[8] Domingos, P. and Hulten, G., "Mining High-Speed Data Streams," *Proceedings of the 6th ACM International Conference on Knowledge Discovery and Data Mining,* pp. 71–80 (2000).

[9] Maron, O. and Moore, A., "Hoeffding Races: Accelerating Model Selection Search for Classification and Function Approximation," *Advances in Neural Information Processing Systems,* pp. 59–66 (1993).

[10] Gama, J., Rocha, R. and Medas, P., "Accurate Decision Trees for Mining High-Speed Data Streams,"

Proceedings of the 9th ACM International conference on Knowledge discovery and data mining, pp. 523–528 (2001).

[11] Hulten, G., Spencer, L. and Ddmingos, P., "Mining Time-Changing Data Streams," *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* pp. 97−106 (2002).

[12] Verykios, V. S., Bertino, K., Fovino, I. N., Provenza, L. P., Saygin, Y. and Theodoridis, Y., "State-of-the-Art in Privacy Preserving Data Mining," *ACM SIGMOD Record,* Vol. 33, pp. 50–57 (2004).

[13] Du, W. and Zhan, Z., "Building Decision Tree Classifier on Private Data," *Proceedings of IEEE International Conference on Privacy Security and Data Mining,* pp. 1–8 (2002).

[14] Agrawal, R. and Srikant, R., "Privacy-Preserving Data Mining," *Proceedings of ACM SIGMOD International Conference on Management of Data,* pp. 439–450 (2000).

[15] Johnsten, T. and Raghavan, V., "Security Procedures for Classification Mining Algorithms," *Proceedings of the 15th Annual Working Conference on Database and Application Security,* pp. 285–297 (2001).

[16] Meregu, S. and Ghosh, J., "Privacy-Preserving Distributed Clustering Using Generative Models," *Proceedings of the 3rd IEEE International Conference on Data Mining,* pp. 211–218 (2003).

[17] Oliveira, S. R. and Zaiane, O. R., "Protecting Sensitive Knowledge by Data Sanitization," *Proceedings of the 3rd IEEE International Conference on Data Mining,* pp. 613–616 (2003).

[18] Lee, G., Chang, C. Y. and Chen, A. L. P., "Hiding Sensitive Patterns in Association Rules Mining," *Proceedings of the 28th IEEE International Conference on Computer Software and Applications,* pp. 424–429 (2004).

[19] Kantarcioglu, M. and Clifton, C., "Privacy-Preserving Distributed Mining of Association Rules on Horizontally Partitioned Data," *IEEE Transactions on Knowledge and Data Engineering,* Vol. 16, pp. 1026–1037 (2004).

[20] Traub, J. F., Yemini, Y. and Wozniakowski, H., "The Statistical Security of a Statistical Database," *ACM Transaction Database Systems,* Vol. 9, pp. 672–679 (1984).

[21] Adam, N. R. and Wortmann, J. C., "Security-Control Methods for Statistical Databases: A Comparative Study," *ACM Computing Surveys,* Vol. 21, pp. 515–556 (1989).

[22] Aggarwal, C. C. and Yu, P. S., "A Condensation Approach to Privacy Preserving Data Mining," *Proceedings of the 9th International Conference on Extending Database,* pp. 183–199 (2004).

[23] Domingo-Ferrer, J. and Torra, V., "Ordinal, Continuous and Heterogeneous k-Anonymity through Microaggregation," *Proceedings of International Conference on Data Mining and Knowledge Discovery,* pp. 195–212 (2005).

Manuscript Received: Jan. 3, 2008
Accepted: Apr. 1, 2009