Mr. Tusharkumar J. Trambadiya, Mr. Praveen bhanodia / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 3, Issue 1, January -February 2013, pp.1096-1103

A Heuristic Approach to Preserve Privacy in Stream Data with Classification

Mr. Tusharkumar J. Trambadiya*, Mr. Praveen bhanodia**
*(Department of Computer Science Engineering, Rajiv Gandhi Prodyogiki Vishwavidyalaya, Indore, India.)
** (Department of Computer Science Engineering, Rajiv Gandhi Prodyogiki Vishwavidyalaya, Indore, India)

ABSTRACT

Data stream Mining is new era in data mining field. Numerous algorithms are used to extract knowledge and classify stream data. Data stream mining gives birth to a problem threat of data privacy. Traditional algorithms are not appropriate for stream data due to large scale. To build classification model for large scale also required some time constraints which is not fulfilled by traditional algorithms. In this Paper we propose a Heuristic approach to preserve privacy with classification for stream data. This approach preserves privacy and also improves process to extract knowledge and build classification model for stream data. This method is implemented in two phases. First is processing of data and second classification analysis. In these two phases first data stream perturbation is applied on data set and after that classification is applied on perturbed data as well as original dataset. Experimental results and charts show that this approach not only preserve privacy but it can also reduces complexity to mine large scale stream data.

Keywords - **Classification, Data Mining, Data Perturbation, Hoefffiding tree, Privacy Preserving, Stream data**

I. 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. Data mining techniques are suitable for simple and structured data sets like relational databases, transactional databases and data warehouses. Fast and continuous development of advanced database systems, data collection technologies, and the World Wide Web, makes data grow rapidly in various and complex forms such as semi structured and non-structured data, spatial and temporal data, and hypertext and multimedia data. Therefore, mining of such complex data becomes an important task in data mining realm. In recent years different approaches are proposed to overcome the challenges of storing and processing of fast and continuous streams of data [2, 3].

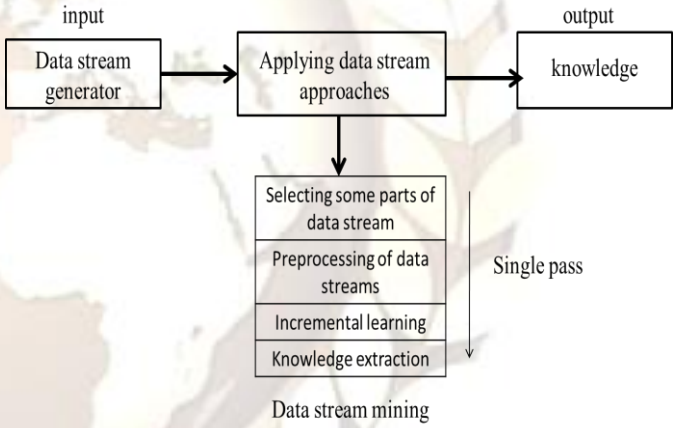

Fig 1.1 General Process of data stream mining

Data stream can be conceived as a continuous and changing sequence of data that continuously arrive at a system to store or process [4]. Imagine a satellite-mounted remote sensor that is constantly generating data. The data are massive (e.g., terabytes in volume), temporally ordered, fast changing, and potentially infinite. These features cause challenging problems in data streams field. Data Stream mining refers to informational structure extraction as models and patterns from continuous data streams. Data Streams have different challenges in many aspects, such as computational, storage, querying and mining. Based on last researches, because of data stream requirements, it is necessary to design new techniques to replace the old ones. Traditional methods would require the data to be first stored and then processed off-line using complex algorithms that

make several pass over the data, but data stream is infinite and data generates with high rates, so it is impossible to store it. [5, 6]:

In, Traditional data mining techniques usually require Entire data set to be present, random access (or multiple passes) to the data, much time per data item. But there are some Challenges of stream mining that are Impractical to store the whole data, Random access is expensive, simple calculation per data due to time and space constraints.

Mining data streams is concerned with extracting knowledge structures represented in models and patterns in non-stopping streams of information. The general process of data stream mining is depicted in Fig. 1.1[7].

II. DATA STREAM RELATED WORK

A classification algorithm must meet several requirements in order to work with the assumptions and be suitable for learning from data streams. The requirements, numbered 1 through 4, are listed below. Also show in figure 2.1 [8]

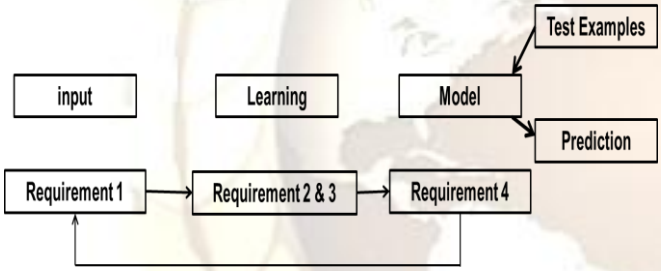

Fig.2.1 Data stream classification cycle

1. Process an example at a time, and inspect it only once (at most)
2. Use a limited amount of memory
3. Work in a limited amount of time
4. Be ready to predict at any point

Figure 2.1 illustrates the typical use of a data stream classification algorithm, and how the requirements fit in. The general model of data stream classification follows these three steps in a repeating cycle: [8]

1. The algorithm is passed the next available example from the stream (requirement1).
2. The algorithm processes the example, updating its data structures. It does so without exceeding the memory bounds set on it (requirement 2), and as quickly as possible (requirement 3).

The algorithm is ready to accept the next example. On request it is able to supply a model that can be used to predict the class of unseen examples (requirement 4).

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 [9].

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.

1. The first category is learning without keeping instances [10]. 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.

2. The second category is learning with partial instance memory. Maloof and Michalski [11] 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.

3. The third category is learning with complete instances [12]. 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 [13] developed a streaming ensemble algorithm for classification. First, the algorithm splits data into several fix sized 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. Ddmingos and Hulten [14] 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 [15] 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.

III. PROBLEM STATEMENT

The goal is to transform a given data set into modified version that satisfies a given privacy requirement and preserves as much information as possible for the intended data analysis task. We can compare the classification characteristics in terms of less information loss, response time, and more privacy gain so get better accuracy of different data stream algorithms against each other and with respect to the following benchmarks:

- Original, the result of inducing the classifier on unperturbed training data without randomization.
- Perturbed, the result of inducing the classier on perturbed data (Perturbation based methods for privacy preserving perturb individual data values or the results of queries by swapping, condensation, or adding noise.) but without making any corrections for perturbed. Show the graphically represent of above defined work in figure.3.1.

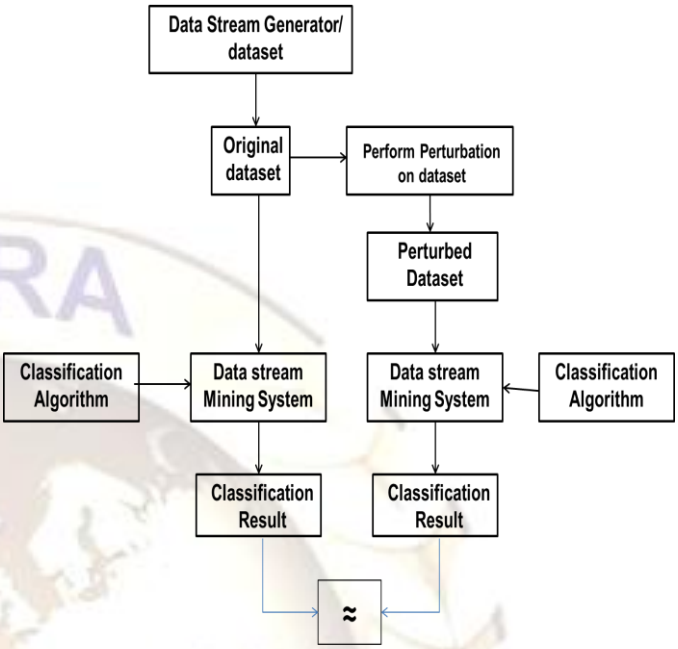

Fig 3.1: Framework for privacy preserving in data stream classification

IV. PROPOSED SOLUTION

Data perturbation refers to a data transformation process typically performed by the data owners before publishing their data. The goal of performing such data transformation is two-fold. On one hand, the data owners want to change the data in a certain way in order to disguise the sensitive information contained in the published datasets, and on the other hand, the data owners want the transformation to best preserve those domain-specific data properties that are critical for building meaningful data mining models, thus maintaining mining task specific data utility of the published datasets [16]. The stage of data streams pre-processing uses 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.

Algorithm: Data Perturbation
Input: An Original Dataset D_S (Original Dataset)
Output: A perturbed Dataset D_S' (perturb dataset)
Algorithm Step:
1) Read Original Dataset $\mathbf{D_S}$ file.
2) Select sensitive attribute among the entire numeric attribute.
3) If selected attribute is **S** then

a. Assign window to S attribute (window store received dataset according to order of arrival)
b. Suppose size of sliding window is ***w***(selection of size at run time) than it contains only ***w*** tuples of selected attribute S values
c. Find mean of ***w*** tuples per window.
d. Replace first tuple of window by mean that we find in above step 3c.
e. Remaining tuples remain as it is.
f. Apply window means remove first tuple of window and insert next tuple from original dataset to end of window. So sliding window size remains same.

4) Again find mean of modified window so go to step 3(a) to 3(f) until all the values of attribute S is changed.

Then modified dataset save in .CSV or .ARFF file. Also called perturbed dataset **D_S'.**

***Description about algorithm*:**

See following **sample** table 4.1 original dataset S and table 4.2 Perturbation dataset D_S' in this table selected attribute S is **Salary**, so compare both table salary attribute those contain salary attribute original data and perturbed data.

TABLE 4.1
Original Dataset D_S, before applying Algorithm

Record No.	Age	Education level	Salary
1	23	15	**53**
2	31	14	**55**
3	33	18	**62**
4	36	11	**49**
5	42	15	**63**
6	48	18	**70**

TABLE 4.2
Perturbation dataset D_S', After apply algorithm

Record No.	Age	Education level	Salary
1	23	15	**57.5**
2	31	14	**52.0**
3	33	18	**57.5**
4	36	11	**52.0**
5	42	15	**62.0**
6	48	18	**71.5**

See Figure: 4.1 and following step 1 to step 3 are for basic concept of sliding window.

1. Apply window to selected attribute S (here selected attribute is Attribute1 contain window size is 6 Tuple1 to tuple6).
2. Find mean of attribute1 (mean of tuple1 to tuple6) and replace first value of window that is 12 by mean. And remaining tuple values are as it is.

Mean is 34.83 than replace first value from window by mean values 34.83 and remaining as it is. And slide window by 1 tuple so new window is from tuple2 to tuple7. Then again find modified wind mean and replace until all values of attribute1 is change. See figure 4.2

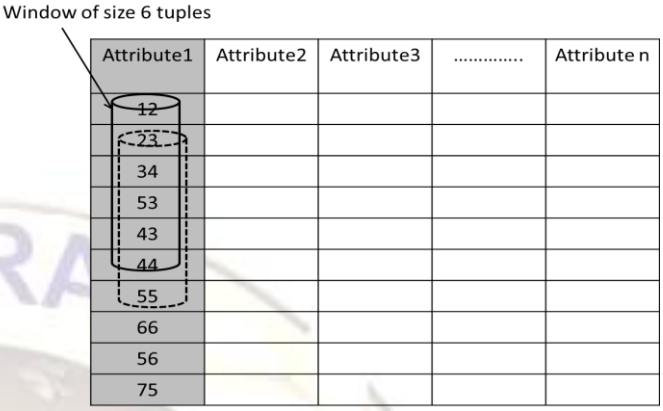

Fig 4.1: window concept (1)

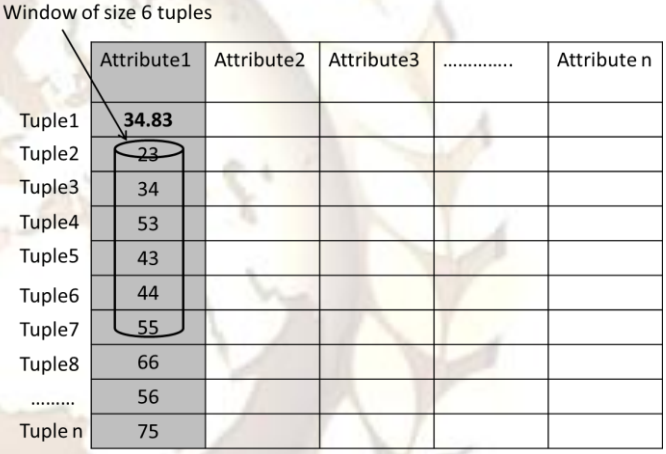

Fig 4.2: window concept (2)

V. EXPERIMENTAL SETUP AND RESULT

We have conducted experiments to evaluate the performance of data perturbation method. We choose generated Database. Generate a dataset from Massive Online Analysis (MOA) Framework [24,25]. And use the Agrawal dataset generator. We use Waikato Environment for Knowledge Analysis (WEKA) [26] tool that is integrated with MOA to test the accuracy of *Hoeffding tree algorithm*. The data perturbation algorithm implemented by a separate Java program.

Following are the basic step for how to perform whole experiment.

Step1. Generate a dataset or take a dataset. In this step we are generate the dataset from MOA generator or take a dataset from UCI data repository.

Step2. Apply the algorithm on dataset and generate perturbed dataset. In this step we apply the

algorithm on dataset. (*Perturb The Data Using Window approach*)

Step3. Take one classification algorithm (*Hoeffding tree*) and apply on perturbed dataset. Use WEKA (MOA integrated) tool

Step4. Generate a Classification model.

Agrawal dataset: Agrawal dataset that is generated by using MOA Framework that contain 200000 instances and 10 attributes. ***Generator.AgrawalGenerator*** [20] Generates one of ten different pre-defined loan functions. It was introduced by Agrawal et al. in [20]. It was a common source of data for early work on scaling up decision tree learners. The generator produces a stream containing nine attributes, six numeric and three categorical. Although not explicitly stated by the authors, a sensible conclusion is that these attributes describe hypothetical loan applications. There are ten functions defined for generating binary class labels from the attributes. Presumably these determine whether the loan should be approved.

Adult Dataset: Extraction was done by Barry Becker from the 1994 Census database. A set of reasonably clean records was extracted using the following conditions: ((AAGE>16) && (AGI>100) && (AFNLWGT>1) && (HRSWK>0)). Prediction task is to determine whether a person makes over 50K a year. Adult dataset contain 48842 instances and 14 attributes. [21]

See the table 7.1 to 7.5 for experimental result of taken three original dataset also contain the result of perturbed dataset that generated by using perturbation algorithm. In table we denote **w** as a window size for data perturbation using window concept. In the first step we generate a perturbed dataset from taken original dataset by using proposed both perturbation algorithm that is *Perturb the Data Using Window approach*. By using Perturbation and Window approach we generate two perturbed dataset and window size is 3 and 4. In second step we apply the *Hoeffding Tree Algorithm* [19] with parameter Split Criterion are Info Gain**,** Tie Threshold: 0.05**,** Split Confidence: 0 on original dataset as well as perturbed dataset and create a classification model.

TABLE 5.1
Experimental result of Agrawal dataset

	Original Dataset	**Perturbed Dataset (window approach)**	
	Agrawal Dataset	**w=3**	**w=4**
Instances	200000		
Attribute	10 (6 numeric)		
Time for model create(s)	0.59	0.51	0.83
Correctly classifier (%)	95.09	67.25	67.27
Kappa statistic	0.8885	0	0

TABLE 5.2
Experimental result of Adult dataset

	Original Dataset	**Perturbed Dataset (sliding window)**	
	Adult Dataset	**w=3**	**w=4**
Instances	32561		
Attribute	14 (6 numeric)		
Time for model create(s)	0.09	0.09	0.08
Correctly classifier (%)	82.52	81.43	81.47
Kappa statistic	0.4661	0.3982	0.406

TABLE 5.3
Confusion matrix of Agrawal Dataset

	Original	
Class	**Yes**	**No**
Yes	129712	4860
No	4954	60747
%	95.09	

TABLE 5.4
Confusion matrix of Adult Dataset

	Original	
Class	**Yes**	**No**
Yes	3791	4050
No	1639	23081
%	82.52	

TABLE 5.5
Confusion matrix of Perturbed Agrawal Dataset

	Perturbed Dataset			
	w=3		w=4	
Class	**Yes**	**No**	**Yes**	**No**
Yes	134445	127	134530	42
No	65366	62	65406	22
%	67.25		67.27	

TABLE 5.6
Confusion matrix of Perturbed Adult Dataset

	Perturbed Dataset			
	w=3		w=3	
Class	**Yes**	**Yes**	**Yes**	**Yes**
Yes	3016	3016	3016	3016
No	1213	1213	1213	1213
%	81.43		81.47	

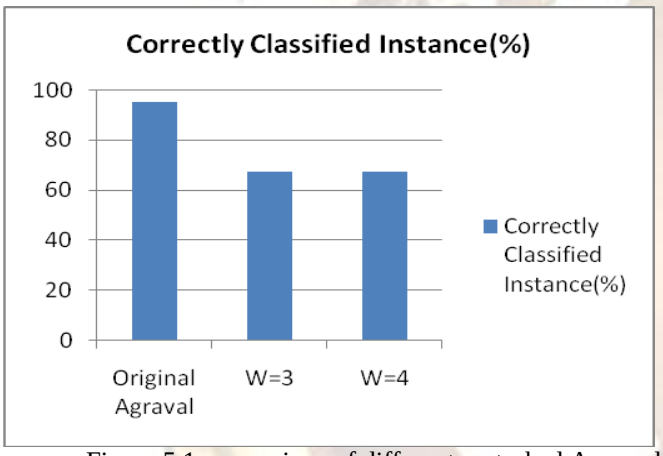

Figure 5.1 comparison of different perturbed Agrawal dataset classification model (in terms of correctly classified instances)

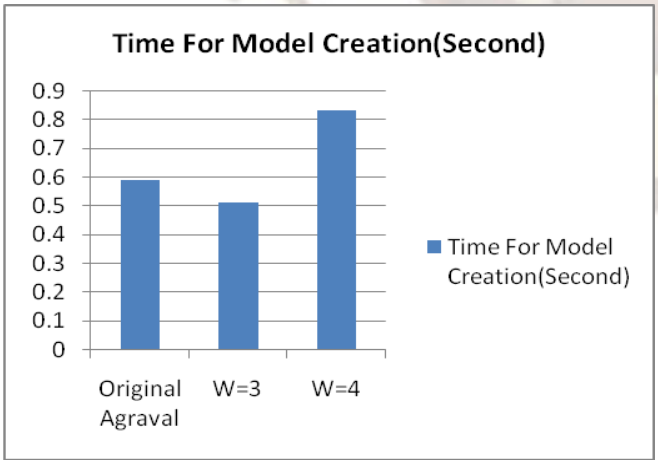

Figure 5.2 Time taken to build Model (Agrawal dataset)

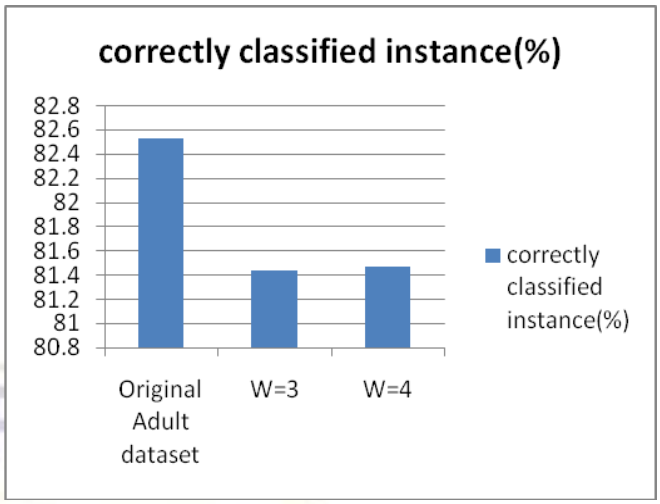

Figure 5.3 comparison of different perturbed Adult dataset classification model (in terms of correctly classified instances)

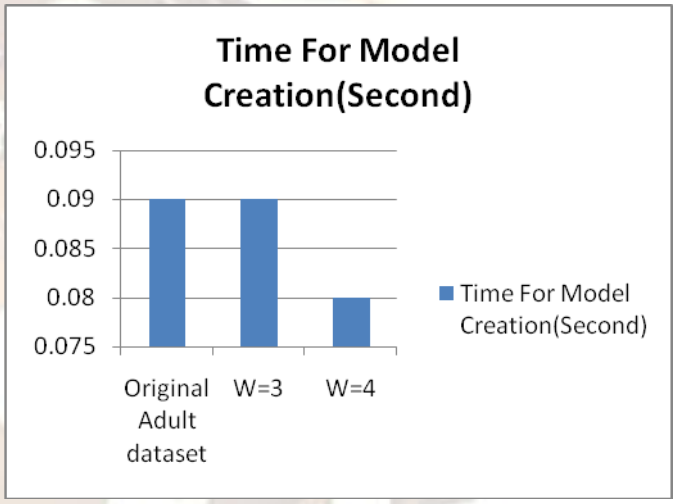

Figure 5.4 Time taken to build Model (Adult dataset)

Misclassification Error:

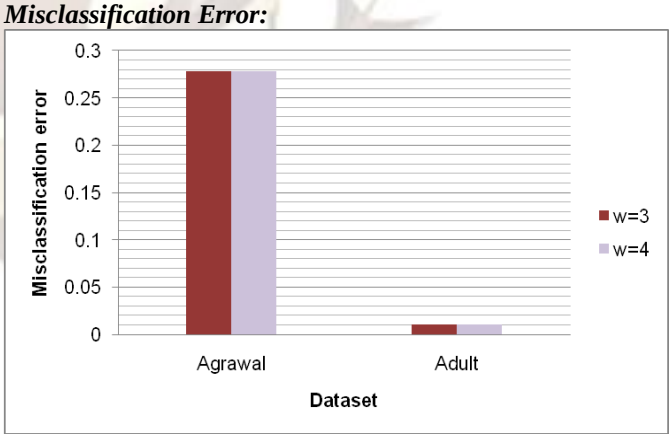

Figure 5.5 Misclassification error

Table 5.7 Experimental result in terms of misclassification error

Dataset	Agrawal	Adult
w=3	0.2783	0.0109
w=4	0.2781	0.0105

VI. CONCLUSION

In this implementation paper we proposed the data perturbation method for privacy-preserving classification of data streams, which consists of two steps: date streams preprocessing and data streams mining. In the step of data streams preprocessing, we proposed algorithm for data perturbation that is used for perturb the data using window approach algorithm. Perturbation techniques are often evaluated with two basic metrics: level of privacy guarantee and level of model-specific data utility preserved, which is often measured by the loss of accuracy for data classification. By using data perturbation algorithm we generate different perturbed dataset. And in the second step we apply the Hoeffding tree algorithm on perturbed dataset. We have done experiment to generate classification model of original dataset and perturbed dataset. We evaluate the experiment result in terms of correctly classified instance, Misclassification error, kappa statistic, and data error measurement. The classification result of perturb dataset shows minimal information loss from original dataset classification.

REFERENCES

[1] D. Hand, H. Mannila and P. Smyth, "Principles of Data Mining", The MIT Press,2001
[2] Babcock, B. Babu, S. Datar, M. Motwani, R., and Widom, J.,"Models and issues in data stream systems". In Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems (PODS). Madison, Wisconsin, pp. 1-16, 2002.
[3] Muthukrishnan, "Data streams: algorithms and applications", In Proceedings of the fourteenth annual ACM-SIAM symposium on discrete algorithms, 2003.
[4] Chaudhry, N.A., Show, K., and Abdelgurefi, M. "Stream Data Management", Advances in Database system, Springer, Vol. 30, 2005.
[5] Aggarwal, C.C, "Data Streams: Models and Algorithms", Springer, 2007.
[6] Chu, F,"Mining Techniques for Data Streams and Sequences", Doctor of Philosophy Thesis: University of California, 2005.
[7] M. Kholghi and M. Keyvanpour, "An analytical framework for data stream mining techniques based on challenges and requirements", International Journal of engineering science and technology (IJEST).
[8] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer, "MOA: Massive Online Analysis", Journal of Machine Learning Research (JMLR), 2010.
[9] Utgoff, P. E., "Incremental Induction of Decision Trees", Machine Learning, Vol. 4, pp. 161-186, 1989.
[10] 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.
[11] Maloof, M. A. and Michalski, R. S., "Incremental Learning with Partial Instance Memory", Foundations of Intelligent Systems, Vol. 2366, pp. 16-27, 2002.
[12] Jin, R. and Agrawal, "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.
[13] Street. 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.
[14] Ddmingos, 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.
[15] Maron, O. and Moore, "Hoeffding Races: Accelerating Model Selection Search for Classification and Function Approximation", Advances in Neural Information Processing Systems, pp. 59-66, 1993.
[16] 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.
[17] 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.
[18] 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.
[19] 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.
[20] Agrawal, R. and Srikant, R., "Privacy-Preserving Data Mining", Proceedings of ACM SIGMOD International Conference on Management of Data, pp. 439-450, 2000.
[21] 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.

[22] Keke Chen, Ling Liu, "A survey of multiplicative perturbation for privacy preserving data mining".

[23] Chen, K., and Liu, L. "A random rotation perturbation approach to privacy preserving data classification", Proc. of Intl. Conf. on Data Mining (ICDM), 2005.

[24] Albert Bifet,Geo olmes,and Bernhard Pfahringer, "MOA: Massive Online Analysis, a Framework for Stream Classification and Clustering", Workshop on Applications of Pattern Analysis, Workshop and Conference Proceedings 11 pp 44-50, 2010.

[25] Albert Bifet, Richard Kirkby,Philipp Kranen, Peter Reutemann, "massive online analysis", manual, 2011.

[26] "The weka Machine Learning Workbench",http://www.cs.waikato.ac.nz/ml/weka.