# An Effective Supervised Streamed Text Classification Approach for Mining Positive and Negative Examples

Safdar Sardar Khan
BUIT Bhopal India

Divakar Singh
Head CSE Deptt. BUIT Bhopal India

## ABSTRACT
Many data mining techniques have been proposed for mining useful patterns in text documents. However, how to effectively use and update discovered patterns is still an open research issue, especially in the field of text mining. This survey paper is based on effective classification of streamed data for text mining by PNLH & one-class classification SVM for text contained audit, we consider the problem of one-class classification of text streams with respect to concept drift where a large volume of documents arrives at a high speed and with change of user interests and data distribution. In this case, only a small number of positively labelled documents is available for training. And text classification without negative examples revisit, by this we propose a labelling heuristic called PNLH to tackle this problem. PNLH aims at extracting high quality positive examples and negative examples from U and our survey can be used on top of any existing classifiers.

## Index Terms
Text mining, text categorization, partially supervised learning, labelling unlabelled data, pattern mining, information filtering.

## 1. INTRODUCTION
Text mining is the discovery of interesting knowledge in text documents. It is a challenging issue to find accurate knowledge (or features) in text documents to help users to find what they want. The problem of *text mining*, i.e. discovering useful knowledge from unstructured or semi structured text, is attracting increasing attention [4][18][19][21][27]. This paper suggests a new framework for text mining based on the integration of *Information Extraction* (IE) and Knowledge Discovery from Databases (KDD), *data mining*. KDD and IE are both topics of significant recent interest. KDD considers the application of statistical and machine-learning methods to discover novel relationships in large relational databases. IE concerns locating specific pieces of data in natural-language documents, thereby extracting structured information from free text [28][29][30][31].

Traditional data mining assumes that the information to be "mined" is already in the form of a relational database. Unfortunately, for many applications, electronic information is only available in the form of free natural-language documents rather than structured databases. Since IE addresses the problem of transforming a corpus of textual documents into a more structured database, the database constructed by an IE module can be provided to the KDD module for further mining of knowledge as illustrated in figure. Information extraction can play an obvious role in text mining as illustrated.

Due to the rapid growth of digital data made available in recent years, knowledge discovery and data mining have attracted a great deal of attention with an imminent need for turning such data into useful information and knowledge. Many applications, such as market analysis and business management, can benefit by the use of the information and knowledge extracted from a large amount of data. Knowledge discovery can be viewed as the process of nontrivial extraction of information from large databases, information that is implicitly presented in the data, previously unknown and potentially useful for users. Data mining is therefore an essential step in the process of knowledge discovery in databases.

Current research trends on data stream classification are mostly focused on classification measures for fully labelled data streams. However, it is not practical in real life applications, as it is generally too expensive to obtain fully labelled data stream for the limited resources of human power.

Suppose a customer service section of a large enterprise receives thousands of user feedback emails every day. Every day the manager of a section only wants to pay a few minutes to find out the feedback emails of a newly launched product for detailed investigation, hence a text data stream classification system is expected to retrieve all the *ontopic* feedbacks in the incoming email streams. This is also a case for news readers when they are browsing online. Users may only have limited patients to label a part of the *ontopic* text documents from the incoming news stream, and a text classifier is expected to retrieve all the related news stories.

In the problem of one-class classification [13][18][6], a class of objects, called the target class, has to be distinguished from all other objects. The description of the target class should be constructed such that the possibility of accepting objects that are not originated from the target class should be minimized. We propose to integrate one-class classification approaches with streaming data mining, so as to construct a text stream classifier without using negative training samples. The following challenges are identified concept drifting small number of training samples, no negative training samples, noisy data, and limited memory space [33][34].

## 2. RELATED WORK
Many types of text representations have been proposed in the past. A well-known one is the bag of words that uses keywords (terms) as elements in the vector of the feature space. To the best of our knowledge, there is no work so far on one-class classification under data stream scenario. While a few works found have discussed classification of partially labelled data streams. In [22], Wu *et al.* proposed a semi-supervised classifier, which uses a small number of both positive and negative samples to train an initial classifier,

without any further user feedback and training samples. This means that their algorithms cannot cope with concept drift in data streams. Text classifiers using positive and unlabelled example are also discussed in [14][15], where the problem of concept drift is not considered as an issue.

Active learning of data streams is proposed in [3][5][7] and [26], which trains initial classifier on partially labelled data stream, and requires the true label of some certain unlabelled samples for enhancing the classifier. The algorithms in [4] estimate the error of the model on new data without knowing the true class labels. As it is known that concept drift could be caused by changing of user interests, changing of data distribution, or both, the algorithms proposed in cannot cope with concept drift caused by sudden shift of user interests. Their system cannot detect this kind of concept drift without knowing the true class labels. Moreover, in real-life applications, the system is always fed with overwhelming volume of incoming data, which makes it not applicable to require human investigation for the true class label of some unlabelled samples. Learning concept drift from both label and unlabelled text data is proposed in [8] and [21]. The algorithms proposed by Klinkenberg *et al.* in [9] need more than one scan of the dataset, which makes it not applicable for a data stream scenario. Dwi *et al*. focused on expanding the labelled training samples using relevant unlabelled data, so as to "extend the capability of existing concept drift learning algorithms".

Topic tracking [1], a sub-task of topic detection and tracking (TDT), tries to retrieval all *ontopic* news stories from a stream of news stories with a few initial *ontopic* samples, and the is related to our work. For TDT, the concept drift is caused by the evolving of the news story itself. While for our work, the concept drift is caused by changes in the user interests, and/or changes in data distribution. The task of information filtering [12][10] is to classify documents from a stream as either relevant or non-relevant according to a particular user interest with the objective to reduce information load. In adaptive information filtering , a sub-task for information filtering, it is assumed that there is an incoming stream of documents, and that each user interest is represented by a persistent profile. Each incoming document is matched against each profile, and (well-) matched documents are sent to the user. The user is assumed to provide feedback on each document sentto him/her. While for our work, no user feedback is supplied. It is generally believed that ensemble classifier could have better classification accuracy than a single classifier [2]. A range of ensemble classifiers has been proposed by research community [11][ 24][19].

The initial papers [17][20] on classification data streams by ensemble methods use static majority votingand static weighted voting, while the current trends is to use dynamic methods, say, dynamic voting in [11], [24]; dynamic classifier selection in [25] and [19]. And it is concluded in that dynamic methods perform better than static methods. Here, are all dedicated to fully labelled data streams. In [23] and [6], algorithms are proposed to have successfully built text classifiers from positive and unlabelled text documents. We use the same idea discussed in to expand the training data from positive-only to include both positive and negative samples to train base classifiers. And then, ensemble stacking is used to cope with concept drift.

Natural language processing (NLP) is a modern computational technology that can help people to understand the meaning of text documents. For a long time, NLP was struggling for dealing with uncertainties in human languages.

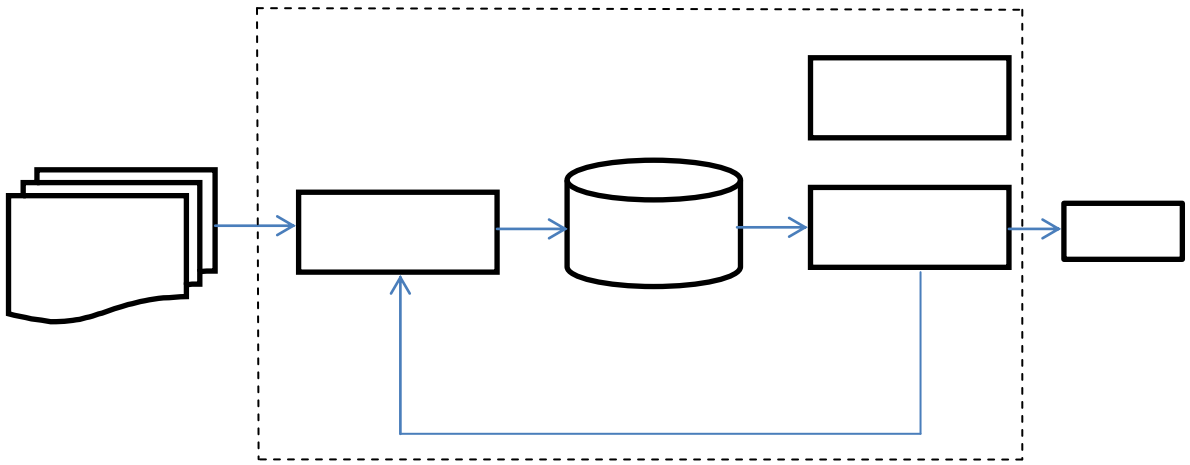

**Figure 1: The architectural view of Text Mining**

## 3. Framework for One-class Text stream Classification

In this paper, we follow the assumption that text stream arrives as batches with variable length [10]:
$d1,1, d1,2, \cdots, d1,m1$ ;
$d2,1, d2,2, \cdots, d2,m2$ ;
$\cdots$ ;
$dn,1, dn,2, \cdots, dn,mn$;

Here, $di,j$ represents the $j$-th document in the $i$-th batch. Each text in the stream is labelled with a category. In each batch, some of the categories are considered as ontopic, and others are not. We write $di,j =< Xi,j, yi,j >$. Here, $Xi,j \in Rn$, representing a sample text in the stream; and $yi,j \in \{+1,-1\}$, representing the document is ontopic ($yi,j = +1$), or not ($yi,j = -1$). In each batch, only a small number of ontopic (positive) samples is given as the training data. The task of one-class text stream classification is to find out all the ontopic documents from the text stream. Algorithm 1 gives a general framework of ensemble learning for one-class text stream classification.

**Algorithm 1** Framework of ensemble learning for one-class data stream classification.
**Input:**
The set of positive samples for current batch, *Pn*;
The set of unlabelled samples for current batch, *Un*;
Ensemble of classifiers on former batches, $En-1$;
**Output:**
Ensemble of classifiers on the current batch, *En*;
1: Extracting the set of reliable negative and/or positive samples *Tn* from *Un* with help of *Pn*;
2: Training ensemble of classifiers *E* on $Tn \cup Pn$, with help of data in former batches;
3: $En = En-1 \cup E$;
4: Classifying samples in $Un - Tn$ by *En*;
5: Deleting some weak classifiers in *En* so as to keep the capacity of *En*;
6: **return** *En*;

It is proved in that the learning task will be more difficult if the learner learns on few samples. Currently, most of the state-of-art classifiers require both positive and negative samples for training. Therefore, in step 1 of algorithm 1, we extract *Tn*, a set of reliable negative samples, from *Un*, the set of unlabelled samples of current batch, so as to create a training dataset, $Tn \cup Pn$. Here, any one-class text classification [23][6] algorithms could be used as a plug-in in this step for extracting samples. In this paper, we simply use a successful algorithm proposed by Fung, *et al.* in [6] for extracting reliable negative samples.

In step 3, the newly learned classifiers are added into $En-1$, the ensemble of classifiers built on former batches of data. In step 4, the new ensemble of classifiers, *En*, is used to determine the class label of unlabelled samples in $Un - Tn$. And in step 5, some weak classifiers are deleted from the ensemble, so as to keep the population capacity of the ensemble.

The overview of the Positive examples and Negative examples Labelling Heuristic (PNLH) is shown in Fig. 2 and Algorithm 2. PNLH consists of two steps: Extraction and Enlargement. The objective of Extraction is to extract a set of reliable negative examples (N) from the unlabelled examples (U) (line 1 of Algorithm 1). The objective of Enlargement is to further extract positive examples (P0) and negative examples (N0) from U _ N (line 3 of Algorithm 2), so as to enlarge P and N.

**Algorithm 2**: PNLH (P, U).
Input: P (positive examples) and U (unlabelled examples)
Output: P (positive training examples) and N (negative training examples)
1. N ←Extract Reliable Negative (P, U);
2. U'← U -N;
3. Obtain P0 and N0 by calling Partition(P, N, U');
4. P ←PÛP';
5. N ←P Û P';
6. return P and N

### *3.1 Extracting Reliable negative Examples*

In the absence of any prior knowledge about the characteristics of the negative examples, the best way to extract a set of negative examples is to use the differences of the feature distributions between the given positive examples (P) and the unlabelled examples (U). In the following, we call the negative documents extracted from U reliable negative examples and denote them by N. We will show how reliable N is in the experimental studies. There are two main procedures in this step, namely, identifying positive features and extracting reliable negative examples.

#### *3.1.1 Identifying Positive Features*

Positive features are the features that frequently appear in P and can represent P well. We identify the positive features based on a notion called core vocabulary. Note that a document that belongs to P must possess some of the features that are contained in the core vocabulary of P, denoted by Vp. According to Vp, we extract the reliable negative examples (N) from the unlabelled examples (U). A document that belongs to the positive examples (P) must possess some of the features that are contained in the core vocabulary of P (VP ).

Algorithm 3: Extract Reliable Negative (P, U).

Input: P (positive examples) and U (unlabelled examples);
Output: N (reliable negative examples);
1. for all fj ϵ P do
2. compute H(fj) using (1);
3. end for
4. compute θ using (2);
5. V← Ɵ;
6. for each fj ϵ P do
7. if H(fj) > θ then
8. Vp ←Vp Û (fj);
9. end if
10. end for
11. compute Ɵ using (5);
12. N ←Ɵ;
13. for all di ϵ U do
14. if G(di) < Ɵthen
15. N← N Ù (di);
16. end if
17. end for
18. return N;

## 4. Experiments

In this section, we report our work on results. The algorithms are implemented in Java NetbeansIDE with help of WEKA1 Software package. Our experiment is made on a PC with i3 2.2 GHz CPU and 2GB memory. 20NewsGroup2 dataset is

used in experiment. The documents are messages posted to Usenet newsgroups, and the categories are the newsgroup themselves. There are 20 categories in this dataset, with almost 1000 text documents for each category. We remove the *subject* header from the text document, as this strongly implies the category of the document. The preprocessing of the text document including stemming and stop words removing. After preprocessing, each document is represented by a vector weighted by TFIDF algorithm. $F1$ is widely used for measuring the classification performance of text classifiers [16]. In this paper, we also report our experiment result in $F1$. Suppose there are $n$ batches of text data observed so far, in order to measure the classification performance on the text stream, we define *averaged* $F1$ for the whole text data stream as the averaged $F1$ over the batches observed from the stream so far, $F1ave = \Sigma i=1\ F1n/n$.

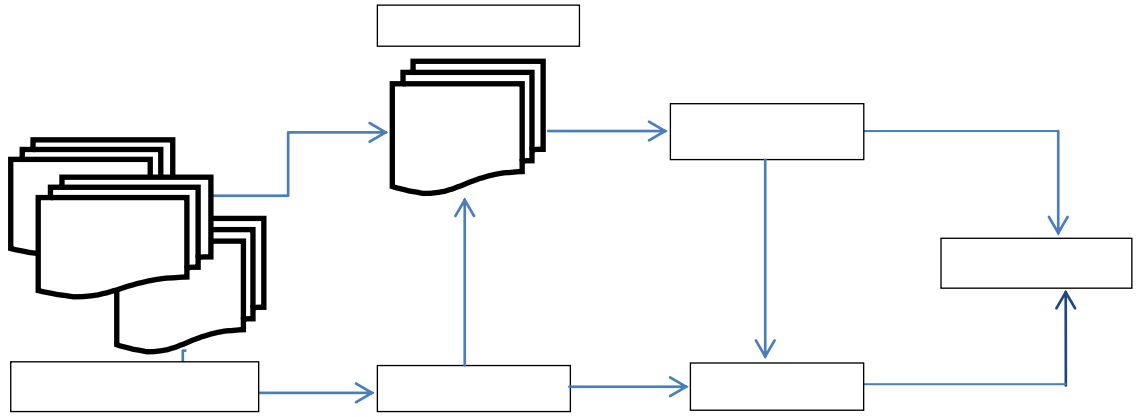

**Figure 2: Identifying resultant classifier by PNLH.**

**Table 5.2: Without PNLH finding accuracy, recall, precision, f1- measure**

| Data sets | With SVM Without PNLH | | | |
|---|---|---|---|---|
| | Accuracy % | Recall % | Precision % | F1-measure % |
| 10 | 0.562 | 0.307 | 0.6 | 0.406 |
| 20 | 0.588 | 0.422 | 0.633 | 0.506 |
| 30 | 0.59 | 0.49 | 0.633 | 0.549 |
| 40 | 0.572 | 0.527 | 0.58 | 0.552 |
| 50 | 0.541 | 0.540 | 0.55 | 0.543 |
| 60 | 0.538 | 0.575 | 0.542 | 0.558 |
| 70 | 0.521 | 0.594 | 0.512 | 0.550 |
| 80 | 0.5 | 0.6 | 0.5 | 0.545 |
| 90 | 0.512 | 0.637 | 0.51 | 0.566 |
| 100 | 0.503 | 0.629 | 0.504 | 0.560 |

**Table 5.3: With PNLH finding accuracy, recall, precision, f1- measure.**

| Datasets | With SVM With PNLH | | | |
|---|---|---|---|---|
| | Accuracy % | Recall % | Precision % | F1-measure % |
| 10 | 0.7 | 0.45 | 0.9 | 0.6 |
| 20 | 0.788 | 0.622 | 0.933 | 0.746 |
| 30 | 0.76 | 0.66 | 0.825 | 0.733 |
| 40 | 0.772 | 0.727 | 0.8 | 0.761 |
| 50 | 0.772 | 0.727 | 0.8 | 0.761 |
| 60 | 0.772 | 0.727 | 0.8 | 0.761 |
| 70 | 0.7 | 0.285 | 0.675 | 0.72 |
| 80 | 0.726 | 0.826 | 0.688 | 0.751 |
| 90 | 0.718 | 0.839 | 0.68 | 0.751 |
| 100 | 0.708 | 0.837 | 0.663 | 0.740 |

The graph shows the result of accuracy, recall, precision and f1-measure without pnlh.

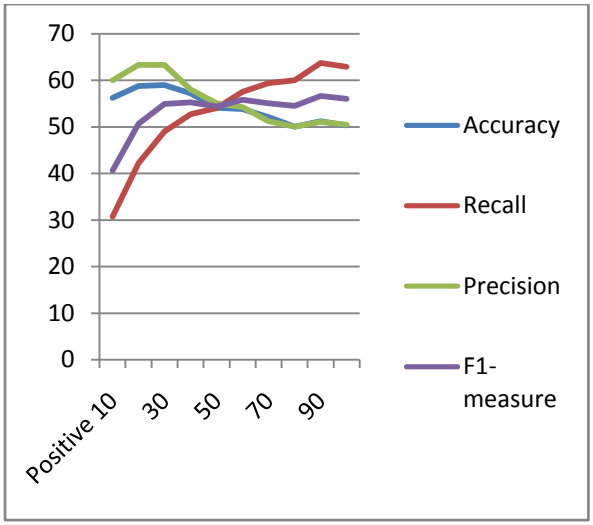


**Figure: The graphical representation of result without pnlh.**

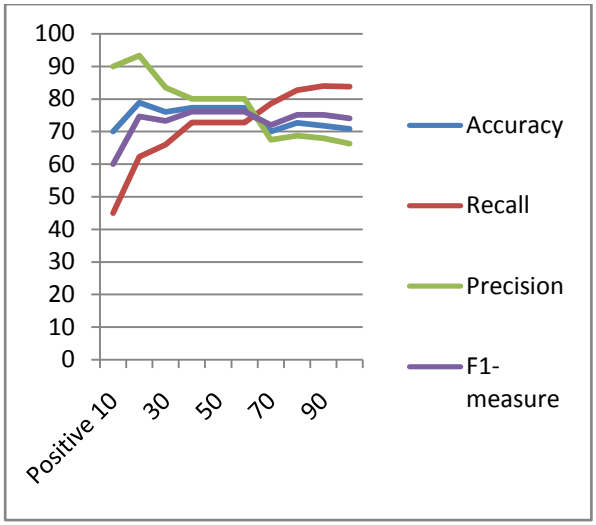


**Figure: The graphical representation of result with pnlh.**

This graph shows the result of accuracy, recall, precision, f1 measure with pnlh.

## 5. CONCLUSION AND FUTURE WORK

Many data mining techniques have been proposed in the last decade. These techniques include association rule mining, frequent item set mining, sequential pattern mining, maximum pattern mining, and closed pattern mining. The main survey of this paper can be summarized as that we firstly tackled the problem of the one-class classification on streaming data. An effective classification of streamed data for text mining by PNLH & one-class SVM for text contained audit. The proposition is that fully labelling streaming data is impractical, expensive, and sometimes unnecessary, especially with text streams. By designing a stacking style ensemble-based classifier, and using a series comparative studies, we have dealt with the problems of concept drift, small number of training examples, no negative examples, noisy data, and limited memory space on streaming data classification. The feature space of text stream may evolve constantly. We need to study the dynamic feature space under the one class text stream classification scenario in the future. On the other hand, the further research should also be considered with the one-class classification on streaming data in general.

## 6. REFERENCES

[1] D.R. Cutting, D.R. Karger, J.O. Pederson, and J.W. Tukey, “Scatter/Gather a Cluster-Based Approach to Browsing Large Document Collections,” Proc. 15th Int’l Conf. Research and Development in Information Retrieval, 1992.

[2] H. Schutze, D.A. Hull, and J.O. Pedersen, “A Comparison of Classifiers and Document Representations for the Routing Problem,” Proc. 18th Int’l Conf. Research and Development in Information Retrieval, 1995.

[3] D. Bennett and A. Demiritz, “Semi-Supervised Support VectorMachines,” Advances in Neural Information Processing Systems,vol. 11, 1998.

[4] P. Bradley and U. Fayyad, “Refining Initial Points for k-Means Clustering,” Proc. 15th Int’l Conf. Machine Learning, 1998.

[5] T. Joachims, “Text Categorization with Support Vector Machines: Learning with Many Relevant Features,” Proc. 10th European Conf. Machine Learning, 1998.

[6] R. Klinkenberg and I. Renz, ”Adaptive information filtering: learning in the presence of concept drifts”. *Workshop Notes of the ICML-98Workshop on Learning for Text Categorization*, pages 33–40, 1998.

[7] B. Larsen and C. Aone, “Fast and Effective Text Mining Using Linear-Time Document Clustering,” Proc. Fifth Int’l Conf. Knowledge Discovery and Data Mining, 1999.

[8] T. Zhang, “The Value of Unlabeled Data for Classification Problems,” Proc. 17th Int’l Conf. Machine Learning, 2000.

[9] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell, “Text Classification from Labeled and Unlabeled Documents Using EM,” Machine Learning, vol. 39, 2000.

[10] R. Klinkenberg and T. Joachims, ”Detecting concept drift with support vector machines,” In Proceedings of the Seventeenth International Conference on Machine Learning (ICML’00), pages 487–494, 2000.

[11] T. Dietterich, “Ensemble methods in machine learning,” Proceedings of the First International Workshop on Multiple Classifier Systems, pages 1–15, 2000.

[12] W. Street and Y. Kim, “A streaming ensemble algorithm (SEA) for large-scale classification,” Proceedings of the seventh international conference on Knowledge discovery and data mining, (KDD’01), pages 377–382, 2001.

[13] D. Tax. One-class classification, ”Doctoral dissertation,” Delft University of Technology, 2001.

[14] Y. Yang, "A Study on Thresholding Strategies for Text Categorization," Proc. 24th Int'l Conf. Research and Development in Information Retrieval, 2001.

[15] J. Allan, "Topic detection and tracking," event-based information organization Kluwer Academic Publishers, 2002.

[16] F. Sebastiani, "Machine learning in automated text categorization," ACM Computing Surveys, vol. 1 pages 1–47, 2002.

[17] J. Bockhorst and M. Craven, "Exploiting Relations Among Concepts to Acquire Weakly Labeled Training Data," Proc. 19th Int'l Conf. Machine Learning, 2002.

[18] R. Ghani, "Combining Labeled and Unlabeled Data for Multiclass Text Categorization," Proc. 19th Int'l Conf. Machine Learning, 2002.

[19] J. Kolter and M. Maloof, "Dynamic weighted majority: a new ensemble method for tracking concept drift," Third International Conference on Data Mining, (ICDM'03), pages 123–130, 2003.

[20] B. Liu, Y. Dai, X. Li, L. W.S., and Y. P., "Building Text Classifiers Using Positive and Unlabeled Examples," Proceedings of the Third IEEE International Conference on Data Mining, (ICDM'03), pages 179–186, 2003.

[21] Page Classification Using SVM," Proc. Ninth Int'l Conf. Knowledge Discovery and Data Mining, 2003.

[22] R. Klinkenberg, "Learning drifting concepts: example selection vs. example weighting," Intelligent Data Analysis, pages 281–300, 2004.

[23] B. Liu, X. Li, L. W.S., and Y. P., "Text Classification by Labeling Words," Proceedings of Nineteeth National Conference on Artificial Intellgience (AAAI-2004), pages 425–430, 2004.

[24] X. Zhu, X. Wu, and Y. Yang, "Dynamic classifier selection for effective mining from noisy data streams," Proceedings of the 4th international conference on Data Mining, (ICDM'04), pages 305–312, 2004.

[25] Symposium on Computer-Based Medical Systems, (CBMS'06), pages 679–684, 2006.

[26] S. Wu, C. Yang, and J. Zhou, "Clustering-training for data stream mining," Sixth IEEE International Conference of Data Mining Workshops, pages 653–656, 2006.

[27] Y. Zhang and X. Jin, "An automatic construction and organization strategy for ensemble learning on data streams," ACM SIGMOD Record, vol. 3, pages 28–33, 2006.

[28] S. Huang and Y. Dong, "An active learning system for mining time-changing data streams," Intelligent Data Analysis, vol. 4, pages 401–419, 2007.

[29] X. Zhu, P. Zhang, X. Lin, and S. Y., "Active Learning from Data Streams," Proceedings of the Sixth International Conference on Data Mining, (ICDM'06), 2007.

[30] X. Jeffrey member, "Text classification without negative examples revist," IEEE computer society 2008.

[31] Z. Zhang Yang, "One-class classification of text streams with concept drift," University of Queensland Australia, 2008.

[32] Z. Jiawei Han and Micheline Kamber, "Data mining concepts and techniques," third edition, 2010.

[33] Arun K Pujari, "Data mining & techniques," second edition, Universities Press, 2011.

[34] Ning Zhong, Yuefeng Li, "Effective Pattern Discovery for Text Mining," IEEE Transactions on Knowledge and data engineering vol.24, No.1 January 2012.