International Journal of Engineering, Science and Mathematics
Vol. 6 Issue 4, August 2017,
ISSN: 2320-0294 Impact Factor: 6.765
Journal Homepage: http://www.ijesm.co.in, Email: ijesmj@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

NOVEL FRAMEWORK FOR DATA STREAMS CLASSIFICATION APPROACH BY DETECTING RECURRING FEATURE CHANGE IN FEATURE EVOLUTION AND FEATURE'S CONTRIBUTION IN CONCEPT DRIFT

Ms. Ritu (Assistant Prof.) BPIT, Delhi

Dr. Bhawna Suri (Associate Prof.) BPIT , Delhi

Dr. P. S. Kulkarni (Assistant Prof.) COER, Roorkee

Abstract:
Data stream classification poses many challenges, most of which are not addressed by the state –of-the-art. These are infinite length, concept drift, concept evolution, feature evolution. Data streams are assumed to be infinite in length, which necessitates single pass incremental learning techniques. Concept drift occurs in a data stream when the underlying concept changes over time. Most existing data stream classification techniques address only the infinite length and concept drift problems. However, to the best of our knowledge, no drift detection method provides insights into which features are involved in the concept drift, which is potentially valuable information. For example, if a feature is contributing to a concept drift it can be assumed that the feature may have become either more or less relevant for the concept encoded in the stream after the drift. This knowledge about a feature's contribution to concept drift could be used to develop an efficient real-time feature selection method that does not require examining the entire feature space for online feature selection. Given a data stream, we want to improve data stream classification accuracy in dynamic feature space by using optimal and dynamic feature selection technique and also address the problem of detecting recurring feature change and feature's contribution in concept drift. This paper proposes a framework for the development of real time feature selection in which we will improve upon the classification accuracy to select the features.

1. Introduction
Data stream classification is the process of finding a general model from past data to apply to new data. Classification is performed in two steps: learning step (training) and testing step. In the learning step, the system tries to learn a model from a collection of data objects, called the training set. In the testing step, the model is used to assign a class label for unlabelled data objects in the testing set. There are four major challenges in data stream classification: infinite length, concept drift , concept evolution and feature evolution. First , we can not store all historical data for training any more because these streams have an infinite length. Second, as more data points arrive in the stream, existing entity distribution patterns may evolve over time which is referred to as concept drift. Therefore in the data stream model, unlike the conventional knowledge discovery tools, we must handle the data stream in limited space and time. A variety of techniques are there for data stream classification such as decision tree, Bayesian classification, support vector machines, k-nearest neighbor, and ensemble classifiers. However, aside from the issues of infinite length and concept drift, another major challenge, feature-evolution, has been ignored by most of the existing works. Feature-evolution occurs because of the feature space that represents a data point in the stream may change over time. For example in a text data stream, new features may become useful and old may become redundant as the stream advances, thus the data's feature space will becomes dynamic. Besides, data dimensions in many application are often very high. To cope with feature-evolution and reduce the computational cost, the classification model should have an efficient unsupervised selection approach due to the absence of class labels.

In high-dimensional data, not all data features (attributes) are important to the learning process. There are three common types of features: (i) irrelevant features, (ii) relevant but redundant features, and (iii) relevant and non-redundant features. The critical task of feature selection techniques is to extract the set of relevant and non-redundant features so that the learning process is more meaningful and faster. Feature selection techniques can be classified into three categories: filter, wrapper, and embedded models . The filter model applies an independent measure to evaluate a feature subset; thus, it only relies on the general characteristics of data. The wrapper model runs together with a learning algorithm and uses its performance to evaluate a feature subset. A hybrid model takes advantage of the above two models. Furthermore, the importance of a feature evolves in data streams and is restricted to a certain period of time. Features that are previously considered as informative may become irrelevant and vice versa; those rejected features may become important features in the future. Thus, dynamic feature selection techniques are required to monitor the evolution of features. Figure 1 illustrates the dynamic nature of key features. Let us suppose that a data stream has three attributes {x,y,z} and two classes: the black and brown dots represent the positive and negative classes, respectively. At timestamp *t*1, the important feature set is {x,y} since data examples are located on the plane {x,y}. Consequently, as the data distribution evolves over time, the key feature set changes to {y,z} at timestamp *t*2.

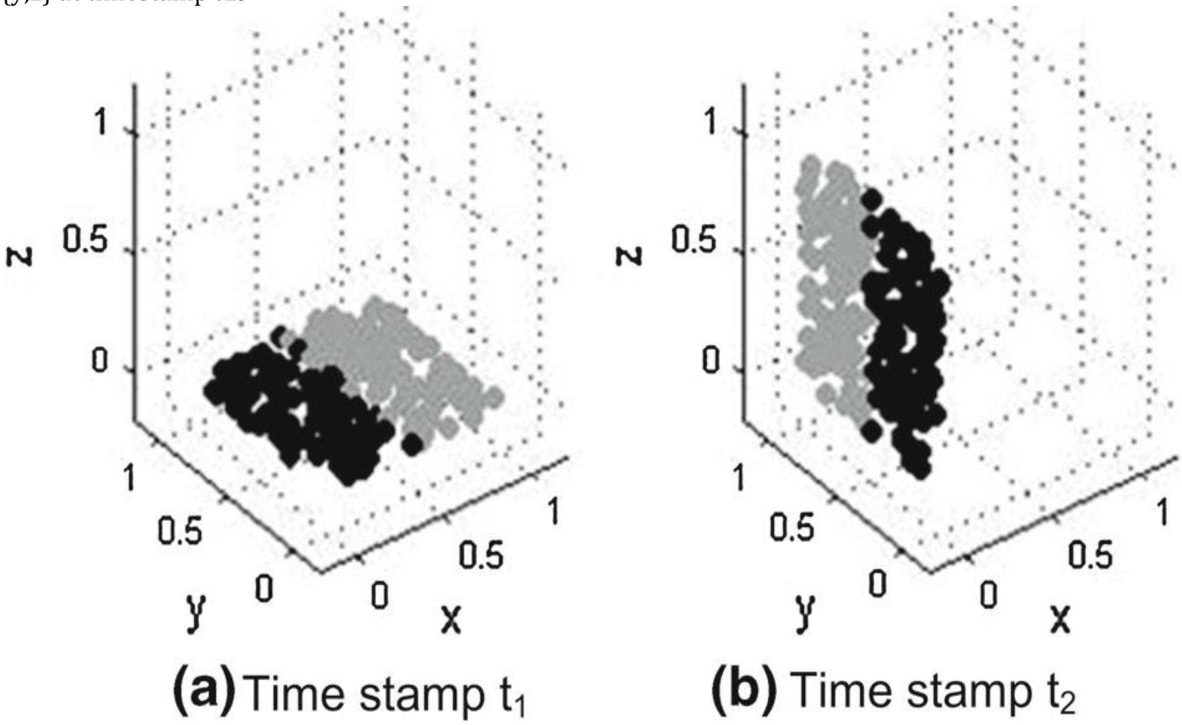

Figure 1 An example showing the dynamic nature of important features. **a** Time stamp *t*1, **b** time stamp *t*2

We have several contributions. First , we will propose a framework for improved classification of data stream which will combine feature selection approaches with concept drift detection module and will also take recurring change in feature/concept in account. Second, to obtain timely and accurate optimal feature subset in dynamic feature space or to reduce the dimensionality of feature space , we will use ideas from matrix sketching to efficiently maintain a low-rank approximation of the observed data and applies regularized regression on this approximation to identify the important features[32]. Thirdly, we will identify feature's importance/contribution in concept drift .

The rest of the paper is organized as follows. Section 2 discusses relevant works in data stream classification. Section 3 describes the proposed framework in detail. Section 4 then explains feature space conversion technique to cope with dynamic feature space. Section 5 will explain contribution of features in concept drift. Finally section 6 concludes the paper.

2. Related work

Most of the existing data stream classification techniques are designed to handle the efficiency and concept-drift aspects of the classification process [2], [1], [20], [23], [11], [8], [5], [6], [7], [24], [21], [4], [13]. Each of these techniques follows some sort of incremental learning approach to tackle the infinite-length and concept-drift problems. There are two variations of this incremental approach. The first approach is a single-model incremental approach, where a single model is dynamically maintained with new data. For example, [8] incrementally updates a decision tree with incoming data, and the method in [2] incrementally updates micro clusters in the model with the new data. The other approach is a hybrid batch-incremental approach, in which each model is built using a batch learning technique. However, older models are replaced by newer models when older models become obsolete [20], [3], [23], [11], [5], [6], [16], [10]. Some of these hybrid approaches use a single model to classify the unlabeled data (e.g., [23]), whereas others use an ensemble of models (e.g., [20], [11]). The advantage of the hybrid approaches over the single model incremental approach is that the hybrid approaches require much simpler operations to update a model (such as removing a model from the ensemble). Our proposed approach not only addresses the infinite length and concept-drift problems but also concept-evolution and feature-evolution. Another category of data-stream classification technique deals with concept-evolution, in addition to addressing infinite-length and concept-drift. Spinosa et al. [19] apply a cluster-based technique to detect novel classes in data streams. Their approach builds a normal model of the data using clustering, defined by the hyper sphere encompassing all the clusters of normal data. This model is continuously updated with stream progression. If any cluster is formed outside this hyper sphere, which satisfies a certain density constraint, then a novel class is declared. However, this approach assumes only one "normal" class, and considers all other classes as "novel." Therefore, it is not directly applicable to multiclass data stream classification, since it corresponds to a "one-class" classifier. Furthermore, this technique assumes that the topological shape of the normal class instances in the feature space is convex. This may not be true in real data. Yet another category of data stream classification techniques address the feature-evolution problem on top of the infinite length and concept-drift problems. Katakis et al. [9] propose a feature selection technique for data streams having dynamic feature space. Their technique consists of an incremental feature ranking method and an incremental learning algorithm. In this approach, whenever a new document arrives belonging to class c, at first it is checked whether there is any new word in the document. If there is a new word, it is added to a vocabulary. After adding all the new words, the vocabulary is scanned and for all words in the vocabulary, statistics (frequency per class) are updated. Based on these updated statistics, new ranking of the words is computed and top N words are selected. The classifier (either kNN or Naive Bayes) is also updated with the top N words. When an unlabeled document is classified, only the selected top N words are considered for class prediction. Wenerstrom and Giraud-Carrier [22] propose a technique, called FAE, which also applies incremental feature selection, but their incremental learner is an ensemble of models. Their approach also maintain a vocabulary. After receiving a new labeled document, the vocabulary is updated, and word statistics are also updated. Based on the new statistics, new ranking of features is computed and top N are selected. Then the algorithm decides, based on some parameters, whether a new classification model is to be created. A new model is created that has only the top N features in its feature vector. Each model in the ensemble are evaluated periodically, and old, obsolete models are discarded often. Classification is done by voting among the ensemble of models. Their approach achieves relatively better performance than the approach of Katakis et al [9]. There are several differences in the way that FAE and our technique approach the feature-evolution problem. First, if a test instance has a different feature space than the classification model, the model uses its own feature space, but the test instance uses only those features that belong to the model's feature space. In other words, FAE uses a Lossy- L conversion, whereas our approach uses Lossless conversion (see Section 4). Furthermore, none of the proposed approaches under this category (feature-evolution) detects recurrent changes in concept and contribution or importance of feature in concept drift.

3. Proposed work

In our paper, we are proposing a framework for feature selection approach that easily adapts to the concept drift arising in data stream by taking the possibility of recurring change in concept/feature in account. By recurring change in feature means that previously used set of features may become useful once again in future. Therefore , it seems promising to develop a method that could accommodate the fact of recurring change in feature selection approach. To the best of my knowledge, no previous work has taken into account the recurring change in features. Previous methods just discard those features which becomes irrelevant for that period of time by selecting top rank features and adjust the model by applying feature space conversion (LOSSY -L, LOSSY-F, LOSSLESS Homogenizing). Simplest way of approaching this would be to create a secondary buffer storing these features for a certain amount of time(by setting some threshold for the frequency of the features and setting the default no of data chunks), allowing to reuse them when necessary. To avoid unacceptable memory requirements, this buffer should be flushed after a certain period of time.

To improve the classification accuracy, in our proposed work, we will use ideas from matrix sketching to efficiently maintain a low-rank approximation of the observed data and applies regularized regression on this approximation to identify the important features. Low rank approximation is a minimization problem in which the cost function measures the fit between a given matrix (the data) and an approximation matrix(the optimization variable). And then, we will apply matrix sketching which is defined as : a sketch of a matrix A is another matrix B which is significantly smaller than A, but

International Journal of Engineering, Science and Mathematics
Vol. 6 Issue 4, August 2017,
ISSN: 2320-0294 Impact Factor: 6.765
Journal Homepage: http://www.ijesm.co.in, Email: ijesmj@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

still approximates it well. As the feature space changes dynamically, so it is very important to filter out the features which are relevant at a particular time .We will apply this on DXminer feature selection approach in dynamic feature space so as to improve classification accuracy.
Proposed framework steps.
Step 1: input- data stream divided into equal fixed size chunks(say N chunks)
Step 2: Train the input stream using ensemble training classification models.
Step 3: Check for prediction accuracy , if it is low than the accuracy for previous chunks,then pass this input stream to concept drift detection module
Step4: If there is a change in the concept then check the contribution of existing features in feature space
Step5: If contribution of any feature is less than threshold, if yes then go to step6 else goto step 7
Step 6: Discard the feature from the feature set temporarily and store it in buffer to check for recurring change in feature(by checking frequency of feature after some constant inputs of data chunks)
Step7: if feature in the buffer is recurring feature , then include it again in feature set, otherwise proceed with step 8
Step8:Apply feature selection module(including feature space conversion module also) to get optimal feature subset.
Step7 output-optimal feature subset with high classification accuracy.

4. Feature space conversion in dynamic feature space

It is obvious that the data streams that do not have any fixed feature space (such as text stream) will have different feature spaces for different models in the ensemble, since different sets of features would likely be selected for different chunks. Besides, the feature space of test instances is also likely to be different from the feature space of the classification models. Therefore, when we need to classify an instance, we need to come up with a homogeneous feature space for the model and the test instances. There are three possible alternatives: i) Lossy fixed conversion (or *Lossy-F* conversion in short), ii) Lossy local conversion (or *Lossy- L* conversion in short), and iii) Lossless homogenizing conversion (or *Lossless* conversion in short).[29]

4.1 Lossy Fixed (Lossy-F) Conversion

Here we use the same feature set for the entire stream, which had been selected for the first data chunk (or first *n* data chunks). This will make the feature set fixed, and therefore all the instances in the stream, whether training or testing, will be mapped to this feature set. We call this a lossy conversion because future models and instances may lose important features due to this conversion. Example: let *FS* = *{Fa,Fb,Fc}* be the features selected in the first *n* chunks of the stream. With the Lossy-F conversion, all future instances will be mapped to this feature set. That is, suppose the set of features for a future instance *x* be: *{Fa,Fc,Fd,Fe}*, and the corresponding feature values of *x* be: *{xa, xc, xd, xe}*. Then after conversion, *x* will be represented by the following values: *{xa,* 0*, xc}*. In other words, any feature of *x* that is not in *FS* (i.e.,*Fd* and *Fe*) will be discarded, and any feature of *FS* that is not in *x* (i.e., *Fb*) will be assumed to have a zero value. All future models will also be trained using *FS*[29].

4.2 Lossy Local (Lossy-L) Conversion

In this case, each training chunk, as well as the model built from the chunk, will have its own feature set selected using the feature extraction and selection technique. When a test instance is to be classified using a model *Mi*, the model will use its own feature set as the feature set of the test instance. This conversion is also lossy because the test instance might lose important features as a result of this conversion. Example: the same example of section 4.1 is applicable here, if we let *FS* to be the selected feature set for a model *Mi*, and let *x* to be an instance being classified using *Mi*. Note that for the Lossy-F conversion, *FS* is the same over all models, whereas for Lossy-L conversion, *FS* is different for different models[29].

4.3 Lossless Homogenizing (Lossless) Conversion

Here, each model has its own selected set of features. When a test instance *x* is to be classified using a model *Mi*, both the model and the instance will convert their feature sets to the union of their feature sets. We call this conversion "lossless homogenizing" since both the model and the test instance preserve their dimensions (i.e., features), and the converted feature space becomes homogeneous for both the model and the test instance. Therefore, no useful features are lost as a result of the conversion. Example: continuing from the previous example, let *FS* = *{Fa,Fb,Fc}* be the feature set of a model *Mi*, *{Fa,Fc,Fd,Fe}* be the feature set of the test instance *x*, and *{xa, xc, xd, xe}* be the corresponding feature values of *x*. Then after conversion, both *x* and *Mi* will have the following features: *{Fa,Fb,Fc,Fd,Fe}*. Also, *x* will be represented with the following feature values: *{xa,* 0*, xc, xd, xe}*. In other words, all the features of *x* will be included in the converted feature set, and any feature of *FS* that is not in *x* (i.e., *Fb*) will be assumed to be zero.

5 Features contribution in concept drift

Our proposed work will also find the feature's importance/contribution in concept drift as the importance of feature changes dynamically over time due to concept drift, important features may become insignificant and vice versa. Previous work is based on maintaining, at each time step t, a low-rank approximation of all the seen (till time t) data stream. By using a regression analysis on this low rank matrix, we can weigh each feature with an up-to-date importance score and then by applying matrix sketching for feature weighing[12]. In our proposed work, feature selection approach can be limited to examining only those features that have changed their weight in concept drift. We can apply weight trusted evaluation on the features set. Initially weight assigned to each feature will be some random no., then we will reduce /increase the weight of the feature during concept drift by some constant by seeing the future stream(some constant data stream chunks)and if a feature reduced its weight to some threshold , then obviously that feature has reduced its contribution to the classification approach. So, it may further contribute to misclassification. Previous work exploit this idea to reduce the complexity of streaming feature selection algorithm. But in our work , we will find feature's importance/weight in concept drift. To the best of our knowledge, no drift detection method provides insights into which features are involved in the concept drift, which is potentially valuable information. For example, if a feature is contributing to a concept drift it can be assumed that the feature may have become either more or less relevant for the concept encoded in the stream after the drift. This knowledge about a feature's contribution to concept drift could be used to develop an efficient real-time feature selection method that does not require examining the entire feature space for online feature selection.

Conclusion

This paper presents an efficient framework for data stream classification by considering recurring changes in features space and also taking into account feature's contribution in the concept drift. The framework produces an optimal feature subset by reducing the dimensionality of the feature space and improves DXMiner on classification accuracy and reduce false alarm rate. In future, this framework can be implemented on real time data streams to check its effectiveness.

.

References

1] C.C. Aggarwal, "On Classification and Segmentation of Massive Audio Data Streams," Knowledge and Information System, vol. 20, pp. 137-156, July 2009.

[2] C.C. Aggarwal, J. Han, J. Wang, and P.S. Yu, "A Framework for On-Demand Classification of Evolving Data Streams," IEEE Trans. Knowledge and Data Eng., vol. 18, no. 5, pp. 577-589, May 2006.

[3] A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, and R. Gavalda`, "New Ensemble Methods for Evolving Data Streams," Proc. ACM SIGKDD 15th Int'l Conf. Knowledge Discovery and Data Mining, pp. 139-148, 2009.

[4] S. Chen, H. Wang, S. Zhou, and P. Yu, "Stop Chasing Trends: Discovering High Order Models in Evolving Data," Proc. IEEE 24th Int'l Conf. Data Eng. (ICDE), pp. 923-932, 2008.

[5] W. Fan, "Systematic Data Selection to Mine Concept-Drifting Data Streams," Proc. ACM SIGKDD 10th Int'l Conf. Knowledge Discovery and Data Mining, pp. 128-137, 2004.

[6] J. Gao, W. Fan, and J. Han, "On Appropriate Assumptions to Mine Data Streams," Proc. IEEE Seventh Int'l Conf. Data Mining (ICDM), pp. 143-152, 2007.

[7] S. Hashemi, Y. Yang, Z. Mirzamomen, and M. Kangavari, "Adapted One-versus-All Decision Trees for Data Stream Classification," IEEE Trans. Knowledge and Data Eng., vol. 21,no. 5, pp. 624-637, May 2009.

[8] G. Hulten, L. Spencer, and P. Domingos, "Mining Time-Changing Data Streams," Proc. ACM SIGKDD Seventh Int'l Conf. Knowledge Discovery and Data Mining, pp. 97-106, 2001.
MASUD ET AL.: CLASSIFICATION AND ADAPTIVE NOVEL CLASS DETECTION OF FEATURE-EVOLVING DATA STREAMS 1495Fig. 5. Scalability of MCM with number of dimensions.
[9] I. Katakis, G. Tsoumakas, and I. Vlahavas, "Dynamic Feature Space and Incremental Feature Selection for the Classification of Textual Data Streams," Proc. Int'l Workshop Knowledge Discovery from Data Streams (ECML/PKDD), pp. 102-116, 2006.

[10] I. Katakis, G. Tsoumakas, and I. Vlahavas, "Tracking Recurring Contexts Using Ensemble Classifiers: An Application to Email Filtering," Knowledge and Information Systems, vol. 22, pp. 371-391, 2010.

[11] J. Kolter and M. Maloof, "Using Additive Expert Ensembles to Cope with Concept Drift," Proc. 22nd Int'l Conf. Machine Learning (ICML), pp. 449-456, 2005.

[12] D.D. Lewis, Y. Yang, T. Rose, and F. Li, "Rcv1: A New Benchmark Collection for Text Categorization Research," J. Machine Learning Research, vol. 5, pp. 361-397, 2004.

[13] X. Li, P.S. Yu, B. Liu, and S.-K. Ng, "Positive Unlabeled Learning for Data Stream Classification," Proc. Ninth SIAM Int'l Conf. Data Mining (SDM), pp. 257-268, 2009.

[14] M.M. Masud, Q. Chen, J. Gao, L. Khan, J. Han, and B.M. Thuraisingham, "Classification and Novel Class Detection of Data Streams in a Dynamic Feature Space," Proc. European Conf. Machine Learning and Knowledge Discovery in Databases (ECML PKDD), pp. 337-352, 2010.

[15] M.M. Masud, Q. Chen, L. Khan, C. Aggarwal, J. Gao, J. Han, and B.M. Thuraisingham, "Addressing Concept-Evolution in Concept- Drifting Data Streams," Proc. IEEE Int'l Conf. Data Mining (ICDM), pp. 929-934, 2010.

[16] M.M. Masud, J. Gao, L. Khan, J. Han, and B.M. Thuraisingham, "A Practical Approach to Classify Evolving Data Streams: Training with Limited Amount of Labeled Data," Proc. IEEE Eighth Int'l Conf. Data Mining (ICDM), pp. 929-934, 2008.

[17] M.M. Masud, J. Gao, L. Khan, J. Han, and B.M. Thuraisingham, "Integrating Novel Class Detection with Classification for Concept- Drifting Data Streams," Proc. European Conf. Machine Learning and Knowledge Discovery in Databases (ECML PKDD), pp. 79-94, 2009.

[18] M.M. Masud, J. Gao, L. Khan, J. Han, and B.M. Thuraisingham, "Classification and Novel Class Detection in Concept-Drifting Data Streams under Time Constraints," IEEE Trans. Knowledge and Data Eng., vol. 23, no. 6, pp. 859-874, June 2011.

[19] E.J. Spinosa, A.P. de Leon F. de Carvalho, and J. Gama, "Cluster- Based Novel Concept Detection in Data Streams Applied to Intrusion Detection in Computer Networks," Proc. ACM Symp. Applied Computing (SAC), pp. 976-980, 2008.

[20] H. Wang, W. Fan, P.S. Yu, and J. Han, "Mining Concept-Drifting Data Streams Using Ensemble Classifiers," Proc. ACM SIGKDD Ninth Int'l Conf. Knowledge Discovery and Data Mining, pp. 226-235, 2003.

[21] P. Wang, H. Wang, X. Wu, W. Wang, and B. Shi, "A Low- Granularity Classifier for Data Streams with Concept Drifts and Biased Class Distribution," IEEE Trans. Knowledge and Data Eng., vol. 19, no. 9, pp. 1202-1213, Sept. 2007.

[22] B. Wenerstrom and C. Giraud-Carrier, "Temporal Data Mining in Dynamic Feature Spaces," Proc. Sixth Int'l Conf. Data Mining (ICDM), pp. 1141-1145, 2006.

[23] Y. Yang, X. Wu, and X. Zhu, "Combining Proactive and Reactive Predictions for Data Streams," Proc. ACM SIGKDD 11th Int'l Conf. Knowledge Discovery in Data Mining, pp. 710 715, 2005.

[24] P. Zhang, X. Zhu, and L. Guo, "Mining Data Streams with Labeled and Unlabeled Training Examples," Proc. IEEE Ninth Int'l Conf. Data Mining (ICDM), pp. 627-636, 2009.

[25] Nguyen, H.,Woon ,Y., Ng, W.:" A Survey on Data Stream Clustering and Classification",Springer 2014

[26]Gallego,S., Krawczyk,B.,Garcia,S.,Wozniak,M.,Herrera, F.:"A survey on Data Preprocesing for Data Stream Mining:Current Status and Future Directions",Springer 2017.

[27] Tennant, M.,Stahl, F.,Rana,O.Gomes,J.:" Scalable real time classification of data streams with concept drift",*Future Generation Computer Systems,*Elsevier 2017,pp187-199

[28]Marron, D.,Read, J.,Bifet,A.,Navaro,N.:"Data Stream classification using random feature functions and novel method cobibaions",*The journal of System and Software*,Elsevier2016,pp110

[29] Masud, M.M.,Chen, Q.,Gao, J., : "Classification and Novel Class Detection of Data Streams in Dynamic Feature space",*In machine learning and knowledge discovery in Databases*, pages 337-352. Springer, 2010.

[30] Masud, M.M.,Chen, Q.,Aggarwal, C.:"Classification and Adapting Novel Class Detection of Feature Evolving Data Streams", *IEEE Transactions on Knowledge and Data Engineering*, Vol 25, No 7, July 2013.

[31] Wang, L., Shen, L. :"Improved Data Stream Classification with Fast Unsupervised Feature Selection",*In Parallel and Distributed Computing , Applications and Technologies*,IEEE, 2016.

[32]Huang,H.,Yoo,S.,Prasad,S.:"Unsupervised Selection in DataStreams",*CIKM*,ACM,2015

[33] Hammoodi ,M., Stahl ,F., Tennant ,M.,:"Towards Online Concept Drift Detection with Feature Selection for Data Stream Classification",*ECSI*,2016