International Journal of Advanced Information in Engineering Technology (IJAIET)
Vol.5, No.2, Feb 2018 ISSN: 2454-6933

TRENDS IN DATA STREAM CLASSIFICATION

S.Jayanthi.
Professor,
Department of Computer Science and Engineering,
Samskruti College of Engg and Technology,
Ghatkesar – 501 301, Hyderabad, India.
nigilakash@gmail.com

***Abstract* - Over a span of decades, Data mining has been applied as a panacea to process data stored in massive data sets by resolving the obstacles confronted by database technologies. However, in recent years, streams of data have been generated in e-industries, which is impossible to make an analysis on data streams by storing it on a stable storage medium. Data mining algorithms which are trained to analyze the data stored in a static storage medium using multiple scanning become unsuitable as the instant response with constrained resources has become the central concern of online analysis. Data stream mining has emerged with several efficient data streaming algorithms to resolve this issue. Among several tasks of data stream mining, data stream classification has frequently been used in e-industries. Despite the availability of a wide range of approaches in the literature, data stream classification is yet in infancy stage where each approach has its own weakness regardless of its strength. The contexts discussed in this paper strongly emphasize the prominence and eminence of stream classification process and its state of the art.**

Keywords: Data streams, Data stream classification, concept drift, concept evolution, data analysis

I. INTRODUCTION

Data streams are evolving in nature and are emanated from diverse distribution centers at varying intensity. Hence, the data stream classification models are imposed to achieve classification with constrained resources in terms of memory and time. To meet this constraint effectively, data stream models take up different data stream classification strategies. The most widely adopted strategies to achieve data stream classification are namely Synopsis of Historical Data and Divide and Conquer Solution.

Synopsis of Historical Data

Since data streams are continuously evolving, tracking the entire history of the data streams is impossible and insignificant. Several approaches track the synopsis of the prominent data and the significant results obtained in the historical data so to make inferences on this statistical report (Gama *et al*., 2013).

Divide and Conquer Solution

To cope with infinite data streams, several approaches perform classification by segmenting the data streams into equally sized data chunks which is pertinent to the convention of divide and conquer (Masud *et al*., 2009; Brzezinski and Stefanowski, 2014a & 2014b).

Incremental Classification

To analyze the dynamic data streams in online, several approaches incrementally update the data stream model by learning the unobserved instances of upcoming data streams (Muhlbaier *et al*., 2009).

Combine Micro and Macro Clustering

Several attempts have been made on combining micro clusters which are formulated using bottom up hierarchical clustering, and macro clusters which are formulated by combining the micro clusters, together to perform prediction with a low margin of error with effectively compressed data.

Combining Block Based and Incremental Classification

Block based and incremental classification approaches are combined to confront different kinds of concept drifts.

Tilted Time Frame Classification Model

Many real time applications are in need to analyze the recent data at a fast pace and they concern the recent data as the most valuable. The classification approaches track the snapshots of the synopsis collected at different time interval using tilted time frame classification model.

On-line and Off-Line Processing of Data Streams

Real time data streams can be processed by discriminating it into online dependent and independent. That is, the classification process does not independent of online can be done in offline mode and vice versa. This strategy is widely adopted to avoid unnecessary mesh in online processing of data.

Ensemble Based Solution

In classification process, yet no single algorithm is proven as panacea for all kinds of data sets and applications and their efficacy is pertinent to the chosen application and data set of the streaming environment. Hence multiple diverse methods can be combined to utilize the strength of more than one classifier in a single classification problem. The proposed research work is focused on formulating ensemble based solution for data stream classification process.

II. CONCEPT DRIFT

Data streams are generated at a quicker pace in online e-industries and mining these data streams are indeed inevitable to better promote their strategic decisions. Data stream classification is the most frequently used task to analyze the data streams in online mode. Concept drift is a very common issue which occurs due to the changes in data distribution centre of the data streaming environment.

Hence, the underlying classification model needs to be restructured or adapted with respect to the evolving concept drifts to sustain its efficacy. Moreover, Concept drift handling systems should also be able to discriminate concept drifts from outliers which are noise that does not meet with the normal behavior of the system, and to react to concept drifts instantly upon their occurrence with limited resource constraints.

Concept drift is the foremost challenge experienced by almost all data stream mining process that seeks an imminent and acute solution to pep up data stream classification process (Sobolewski and Wozniak, 2013).

A. Types of Concept Drift

Concept drifts are classified into four major categories (Zliobaite, 2010):

- Sudden Concept Drift
- Incremental (Stepwise)
- Gradual Concept Drift
- Recurrent Concept Drift

Sudden Concept Drift

In sudden concept drift, the context or nature of data streams changes abruptly at a certain point in time. The current classification model becomes obsolete and unfit for data stream classification upon the incidence of sudden concept drift. Hence, the classification model needs to be re-taught or restructured from the scratch to cope with this kind of concept drift (Yang *et al*., 2011; Gama *et al*., 2013). This scenario is exemplified in Fig.1(Zliobaite, 2010).

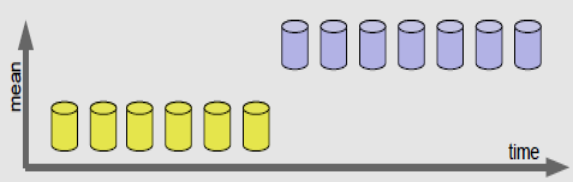
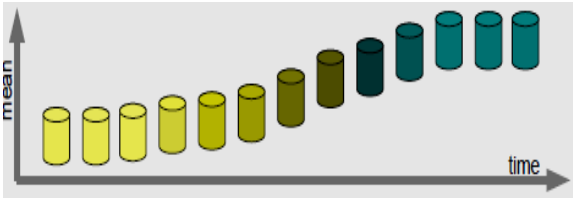


Fig.1. Sudden Concept Drift

Incremental or Stepwise Concept Drift

In incremental concept drift, the context or nature of data streams changes steadily over time. Hence, the classification model needs to be updated incrementally over time. This scenario is exemplified in Fig.2(Zliobaite, 2010).



.

Fig.2. Incremental Concept Drift

Gradual Concept Drift

In gradual concept drift, the context or nature of data streams changes gradually over time. Here, the classification model needs to be updated only upon the incidence of concept drift. This scenario is exemplified in Fig.3(Zliobaite, 2010).

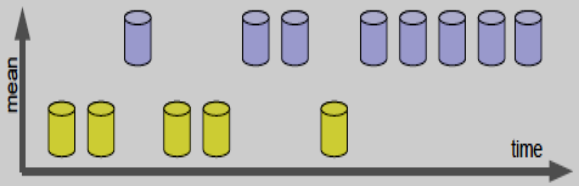


Fig.3. Gradual Concept Drift

Recurrent Concept Drift

In recurrent concept drift, the change in the context or nature of data streams happens regularly in a circular fashion and sets trends over time. The classification model needs to keep track of the significant features of the historical data chunks as snapshots to reduce the cost incurred in relearning of the recurrent concept drift. This scenario is illustrated in Fig.4 (Zliobaite, 2010).

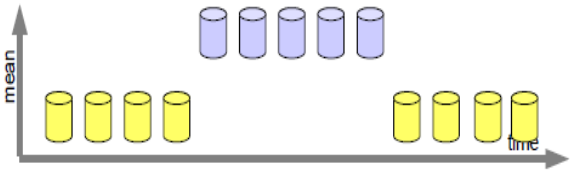


Fig.4. Recurrent Concept Drift

Concept drifts can be handled in any of the following approaches (Sobolewski *et al*., 2013).

- Restructure the classification model, if new data arrive. This approach incurs a high computational cost and especially when concept drifts occur rapidly. So, this method is not widely opted where the possibility for the incidence of concept drift is high.
- Detect changes in the arriving data, and if changes are sufficiently significant, then restructure the classifier.
- Adopt a suitable incremental learning algorithm to update the classification model.

In the proposed classifiers of this thesis, the third approach in the above stated approaches is adopted to devise and exploit the novel incremental learning algorithms to cope with concept drifts in the data stream classification process.

III. CONCEPT DRIFT HANDLING APPROACHES

Incremental algorithms, which updates its classification model with respect to concept drifts, can be classified into four major groups (Zliobaite *et al*., 2010; Brzezinski *et al*., 2014, Stanley *et al*., 2003, Masud *et al*., 2013):

- Online learners
- Instance based solutions
- Drift detection algorithms
- Ensemble approaches

In general, any online learning algorithm deploys the following three steps in an online learning environment. Consider a sequence of input instances x1, x2, ..., xj,... which arrive continuously and infinitely, from an unknown distribution centre D (Masud *et al*., 2009).

In this data streaming scenario, the following sequence is repeated continuously in the classification model:

- The incremental learning algorithm receives unlabeled instances from D.
- The algorithm predicts class labels for each instance of x_n.
- The algorithm is updated with respect to novel instances or concept drifts if any.

The reaction to a concept drift can be made in two ways:

Global Replacement

The entire classification model is restructured from the scratch when the intensity of concept drift is high.

Local Replacement

Only the sub region of the classification model is tuned when the intensity of concept drift level is low.

A. *Online Learners*

On-line learning approaches deploy special data stream learning algorithms which learn incrementally on the incoming data and update their internal hypotheses with respect to each new incoming instance. Upon the arrival of new instances in online, online learners react to concept drifts much faster than in the environment where block based processing takes place (Brzezinski *et al*., 2014; Minku *et al*., 2010; Read *et al*., 2012).

Hoeffding Option Tree (HOT) (Bernhard Pfahringer, 2007) is a regular Hoeffding tree that contains additional *optional* nodes to perform several tests on different attributes, where each test leads to a separate sub tree that needs a controlling mechanism to limit tree growth explosion. Final decisions can be made by combining the weights of the results of subtrees.

However a huge number of incremental learners are available for data stream classification, many attempts have been made to combine online learning and ensemble learning approaches so as to exploit the strength of ensemble classification in online learners. Few of these attempts are discussed in section 3.2.5.

B. *Instance Based Learners*

Instance based learners classify newly arrived unknown instances by finding similarity with the known instances which have been stored in memory. Instance based learners are popularly known as lazy learners. Instance based learners may simply throw the outdated old instances by retaining the recent instances alone. This feature makes them amicable to meet the tight response time and memory constraints of data stream classifier (Gama *et al*., 2004; Read *et al*., 2012; Bifet *et al*., 2013).

K-Nearest Neighbor approach, kernel based approaches, and neural network approaches performs classification based on instance based learning approach.

In general, Euclidean distance is the most preferable choice, used to find similarity between the instances. To state more precisely, let an arbitrary instance x be described by the feature vector (set of attributes) as follows:

$$< a1(x), a2(x), ...an(x) >$$

where *ar(x)* denotes the value of the rth attribute of instance *x*.

Euclidean distance between two instances x_i and x_j is determined by calculating $d(x_i, x_j)$ where

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^{n} (a_r(x_i) - a_r(x_j))^2} \quad (1)$$

However, to measure similarity, a range of other distance metrics, such as Mahalanobis distances (Wei *et al*., 2010), Minkowski distance, etc., are also used in instance based learning classifier.

C. Drift Detection Algorithms

Drift detection algorithms or trigger based algorithms detect and restructure the classification model by comparing the results of actual true class labels with that of the predicted one.

Drift Detection Method (DDM), the most popular drift detection method, calculates the error rate of the classifier at each iteration of the classification process. Classification error rate is modeled using binomial distribution with a constant verification to ensure whether they fall into the bounds of warning level or not. If so, the classifier is replaced with a new classification model constructed with new instances (Gama *et al*., 2004).

Adaptive Classifier Ensemble (ACE), a variant of DDM, is intended to react to sudden concept drifts by tracking the error rate of classifiers for each arriving instances of data streams. ACE extends the validity of classifiers in the ensemble by reconstructing them gradually with respect to large blocks of instances (Nishida, 2008).

Early Drift Detection Method (EDDM), refined version of DDM, detects concept drifts by calculating the distance between two successive classification errors where a significant decrease in the distance indicates concept drifts (Garcia *et al*., 2006).

STEPD discovers concept drift by tracking the predictive accuracy of the proposed classifier using a statistical test. The classifier is reinitialized to the recent concept immediately upon the significant concept drift occurrence (Nishida and Yamauchi, 2007).

D. Ensemble Approaches

The ensemble classifier approach employs diverse classifiers to explicitly handle concept drifts with a change detector by providing a useful description about the drifts. The ensemble classifier based drift detection approach is widely adopted for data stream classification as it is more flexible, robust and accurate in dealing with different types of concept drifts.

The ensemble of classifiers can be articulated by adopting any one of the following ensemble topologies (M.Wozniak *et al*., 2013).

- Vertical Ensemble
- Horizontal Ensemble
- Hybrid Ensemble

In horizontal ensemble approach, classification results of the classifiers in the ensemble are fed as input sequentially to the successive classifiers so as to produce more accurate classification results as shown in Fig.5.

Fig.5. Horizontal Ensemble Classifier

In vertical ensemble approach, instances are fed to more than one classifier in the ensemble simultaneously as shown in Fig.6. Classification results are produced either by adopting majority voting or average weighting approach.

Hybrid ensemble classifier approach exploits the principles of both the vertical as well as the horizontal ensemble classification approach to achieve the strength of both the approaches in the classification process as shown in Fig.7.

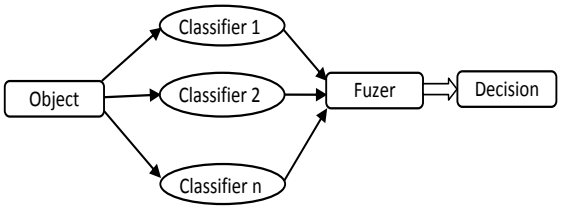


Fig.6. Vertical Ensemble Classifier

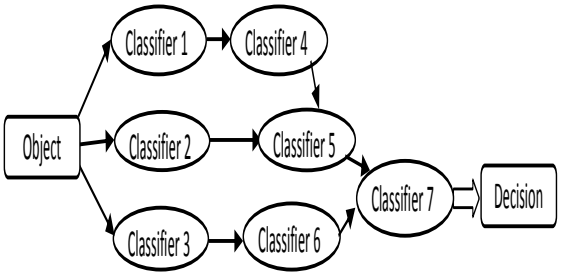


Fig.7. Hybrid Ensemble Classifier

Following are the prominent approaches which exploit ensemble classifier conventions with that of the incremental classifier to achieve online data stream classification. The Streaming Ensemble Algorithm (SEA) is a batch based ensemble

classifier that handles concept drifts by training and adding a new classifier for each batch of new data to the existing ensemble. It prunes the least performing classifier using the simple majority voting approach to limit the size of the ensemble (Street and Kim, 2001). This classifier is efficient in handling sudden and gradual concept, but not good in recurrent concept drift.

The Accuracy Weighted Ensemble (AWE) trains a new classifier on each block of the incoming instance and evaluates all the existing classifiers in the ensemble according to the recent data chunk using a weighted voting approach (Wang.H *et al*., 2003).

Dynamic Weighted Majority (DWM) is an online ensemble classifier approach that adds and removes base learners upon the arrival of new instances with respective to their correctness in local as well as global level, calculated using weighted majority voting approach (Kolter and Maloof, 2003).

Yan-Nei Law *et al*., 2005, proposed an Adaptive Nearest Neighbor Classification Algorithm for Data Streams (ANNCAD) that adaptively finds its nearest neighbouring cells and expands the nearby area of a test point for the arrival of new data points. Its drawback is that it needs additional data structures to confront concept drifts in data stream classification.

Accuracy Updated Ensemble (AUE) (D. Brzezinski *et al*., 2011) is a batch based incremental ensemble approach confronts concept drifts by incrementally selecting and updating the classifiers in the ensemble according to the current distribution. Accuracy Updated Ensemble (AUE2) (D. Brzezinski *et al*., 2014) is an extension of AUE1 that deals with all different kinds of concept drifts with efficacy by adopting accuracy-based weighting approach along with Hoeffding trees.

Weighted Majority Algorithm (WMA) (Littlestone *et al*., 1994) assigns a weight equally to all classifiers when they are newly added to the ensemble. The weight of the classifier is reduced upon their false predictions. It produces the prediction of the classifier, having the highest weight, as the classification result.

Adaptive Size-Hoeffding Tree Bagging (ASHT Bagging) (Albert Bifet, 2009) uses hoeffding trees of various sizes, with the assumption that the small size trees adapt quickly to changes and the largest ones work better for long periods. After each node split, it revises the tree ensemble by deleting some least important nodes to reduce its size.

ADWIN Bagging (Albert Bifet, 2009) uses ADWIN as a change detector and when a drift is detected, it replaces the least efficient classifier with a new one. Online bagging and leverage bagging are the fine tuned version of bagging algorithms which are used widely in ensemble data stream classification.

SmSCluster (M. Masud *et al*., 2008), a semi supervised clustering algorithm that builds micro clusters using semi supervised approach and subsequently performs classification using K-Nearest Neighbour classification algorithm.

Peng Zhang *et al*., 2009, proposed an Aggregate Weighted Ensemble (AWE) Framework, a mixture of horizontal and vertical ensemble classification frameworks which brings forth the strength of both the frameworks to better classify the drifting data streams even in the presence of errors.

IV. ENSEMBLE AGGREGATION STRATEGY

In data streaming environment, data streams are split into several data chunks and scanned separately by several classifiers concurrently. The data chunks from the data stream may be disjoint or overlapping. To tackle this issue, a suitable combining procedure needs to be applied in order to produce the final classification result. This section discusses about the most frequently used aggregation strategies of the ensemble approach.

Boosting

Boosting is a general method used for turning the weak learner of the ensemble into the strong learner by changing the distribution of training instances. The main idea of this algorithm is to assign a weight for each instance in the training set. Initially, equal weight is assigned to all instances of the training set.

At all iterations, the weights of all the misclassified instances are increased (boosted) and the weights of all correctly classified instances are decreased. Subsequently, the weak learner is forced to focus on the weak portions of the dataset.

AdaBoost Algorithm

AdaBoost is a variant of Boosting algorithm which assigns weights for all classifiers and instances. Subsequently, it reduces or increases it weights which are inversely proportional to its resultant classification accuracy.

The formula used to calculate the weight of new instances is given below (Lior Rokach, 2010).

$$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t . Ct(x)\right) \qquad (3.2)$$

T is the total number of iterations. X is the instances of the classifier.

α_t is the weight of the classifier Ct.

Bagging (Bootstrap AGGregation)

The most prominent method aimed to improve the accuracy by creating an improved composite classifier by combining the outputs of various classifiers. It uses the average voting method to

combine the results of the classifiers to produce a single accurate prediction. Bagging is widely adopted in unstable learners, such as decision trees, neural networks, etc. The prediction process in an ensemble is illustrated in Fig.3.8.

V. NOVEL CLASS AND OUTLIERS

Many of the data stream classifiers fix the class labels as stable and do not mind the fact that the data streaming environment is subject to the incidence of novel classes. Data stream classifiers need to discover the incidence of novel classes in the data stream classification process. This section discusses about the various notable approaches available in the literature.

Novel class is the class which is not recognized by the classification model during its training period under existing class labels and may potentially form a novel class with new class labels. Outliers are noise which does not comply with other classified instances and do not have the potential of forming a distinct class.

DXMiner is a data stream classifier which resolves concept evolution or novel class issue by automatically inferring novel classes in data streaming environment. To achieve this, it initially builds a decision boundary around the training data. If novel instances with strong cohesion are found, then they will be declared as novel classes by dynamically changing the feature spaces of the classification models. If sufficient cohesion is not found among the novel instances of data streams, they will be ignored as outliers (Masud *et al*., 2010).

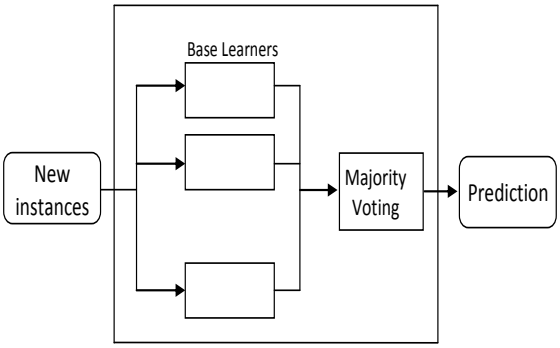


Fig.8. Ensemble prediction process

Masud *et al*. (2011), presented a multiclass framework, ECSMiner that discriminates the instances of novel classes from the instances of existing classes which are stored in buffers containing the summary of clusters, such as, weight, radius and mean distance. It also checks the potentiality of new instances in forming novel classes or outliers, under timing constraints. In the above mentioned research works, distance based metric is used to discriminate novel classes and outliers.

Several attempts have also been made in adopting case based reasoning, rule based reasoning, etc., to achieve the data stream classification task. For instance, Email Classification Using Examples (ECUE) is a data stream classification approach which adopts case based reasoning (Delany, 2006). The data stream classifiers, such as VFDR, AVFDR, etc., adopt the principles of rule based classifiers.

However, convincing results have been attained in a few of the above research works, further research work is needed to strengthen the results of data stream classification process by exploiting suitable strategies to address all the issues of data stream classification.

VI. CONCLUSION

Inferences made through the state of the art of data stream classification are briefed herein this section:

- Concept drift is an inevitable issue that arises in data streams and may take any of the forms such as sudden concept drift, incremental or stepwise concept drift, gradual concept drift, and recurrent concept drift.
- The algorithms which are efficient in handling a particular kind of concept drift are inattentive on other kinds of concept drifts.
- Despite the wide availability of data stream classifiers in the literature, many of them have been not tested on online mode.
- It is also observed that the accuracy of the data stream classifiers has fluctuated over different data streams generated over different scenarios.
- However, many of the algorithms are good at handling concept drifts, they disregard concept evolution and fix the classes in advance.
- Most of the online learning algorithms generate a new classification model at the incidence of sudden concept drift. However it achieves high accuracy, it incurs a high computational cost.
- Many approaches emphasize on resolving any one particular issue of data stream classification, and the panacea for resolving all the issues together in a data streaming application is not found in the literature.

Hearkening to the above limitations inferred in the literature, the novel idea of formulating ensemble classifier based data stream classification has been triggered. In our future research work, it has been planned to investigate the competency and scalability of the ensemble constructed with support vector machine, genetic algorithm, parallel genetic algorithm, Lagrangian interpolation method and K-Means over various

sizes of data chunks generated by a real time server in online mode.

REFERENCES

1. Albert Bifet, Bernhard Pfahringer, Jesse Read, and Geoff Holmes, "Efficient Data Stream Classification via Probabilistic Adaptive Windows", ACM, Proceeding SAC, 2013, pp.801-806.
2. Albert Bifet, Geoff Holmes and Bernhard Pfahringer, "New Ensemble Methods For Evolving Data Streams", Proceeding 15th ACM SIGKDD, 2009, pp. 139-148.
3. Bernhard Pfahringer, Geoffrey Holmes, and Richard Kirkby, "New Options for Hoeffding Trees", Proceeding AI'07, Springer-Verlag Berlin, Heidelberg, 4830, 2007, pp.90-99.
4. Dariusz Brzezinski, and Jerzy Stefanowski,. "Combining block-based and online methods in learning ensembles from concept drifting data streams", Information Sciences, 265, 2014, pp.50–67.
5. Gama, J., Medas, P., Castillo, G., Rodrigues, P, "Learning with drift detection", Lecture Notes in Computer Science, 2004.
6. Haixun Wang, Wei Fan, Philip S. Yu, and Jiawei Han, "Mining concept-drifting data streams using ensemble classifiers", Proceedings of the 9th ACM SIGKDD, 2003, pp. 226–235.
7. Indre Zliobaite, 2010. "Learning under Concept Drift: an Overview", CoRR abs/1010.4784.
8. J. Zico Kolter, and Marcus A. Maloof, "Dynamic Weighted Majority: An Ensemble Method for Drifting Concepts", Journal of Machine Learning Research, 8, 2007, pp. 2755–2790.
9. Jesse Read, Albert Bifet, Geo Holmes, and Bernhard Pfahringer, "Scalable and Efficient Multi-label Classifcation for Evolving Data Streams", Machine Learning, 88, 2012, pp.243-272.
10. Jesse Read, Albert Bifet, Bernhard Pfahringer, and Geo Holmes, "Batch-incremental versus instance-incremental learning in dynamic and evolving data", Proceeding, Springer-Verlag Berlin, 2012, pp. 313-323.
11. Joao Gama, Raquel Sebastiao, and Pedro Pereira Rodrigues, "On evaluating stream learning algorithms, Machine Learning", vol. 90(3), 2013, pp. 317-346.
12. Kenneth O. Stanley, "Learning Concept Drift with a Committee of Decision Trees", Technical Report UT-AI-TR-03-302, 2003.
13. Kyosuke Nishida, "Learning and Detecting Concept Drift, Graduate School of Information Science and Technology", Hokkaido University, Dissertation, 2008.
14. Leandro L. Minku, Allan P. White, and Xin Yao, "The Impact of Diversity on Online Ensemble Learning in the Presence of Concept Drift", IEEE Transactions on Knowledge And Data Engineering, 22(5), 2010, pp.730-742.
15. Littlestone and M. Warmuth, "The weighted majority algorithm", Inf. Comput., 108(2): 1994, pp.212–261.
16. Manuel Baena-Garcia, Jose del Campo-Avila, Raul Fidalgo, Albert Bifet, Ricard Gavalda, and Rafael Morales-Bueno, "Early Drift Detection Method", ECML PKDD'06, 2006.
17. Michael D. Muhlbaier, Apostolos Topalis, and Robi Polikar, "Learn++.NC: Combining Ensemble of Classifiers With Dynamically Weighted Consult-and-Vote for Efficient Incremental Learning of New Classes", IEEE Transactions on Neural Networks, 20(1): 2009, pp.152-168.
18. Mohammad M. Masud, Clay Woolam, Jing Gao, Latifur Khan, Jiawei Han, Kevin W. Hamlen, and Nikunj C. Oza, "Facing the reality of data stream classification: coping with scarcity of labeled data", Knowledge and Information Systems, 2011.
19. Mohammad M. Masud, Jing Gao, Latifur Khan, Jiawei Han, and Bhavani Thuraisingham, "Integrating Novel Class Detection with Classification for Concept-Drifting Data Streams", *ECML PKDD*, Springer-Verlag Berlin Heidelberg, 2009, pp.79–94.
20. Mohammad M. Masud, Member, Latifur Khan, Jiawei Han, and Bhavani Thuraisingham, "Classification and Novel Class Detection in Concept-Drifting Data Streams under Time Constraints", IEEE Transactions on Knowledge And Data Engineering, 23(6), 2010, pp. 859-874.
21. Mohammad M. Masud, Qing Chen, Latifur Khan, Charu C. Aggarwal, Jing Gao, Jiawei Han, Ashok Srivastava, and Nikunj C. Oza, "Classification and Adaptive Novel Class Detection of Feature-Evolving Data Streams", IEEE Transactions On Knowledge And Data Engineering, 25(7): 2013, pp.1484-1497.
22. Peng Zhang, Xingquan Zhu, Yong Shi, and Li Guo, Xindong Wu, "Robust Ensemble Learning for mining noisy data streams", Decision Support System, 2011, pp.469-479.
23. Sarah Jane Delany, "Using Case-Based Reasoning for Spam Filtering", Dublin Institute of Technology, dissertation, 2006.
24. Sobolewski P., Wozniak M., "Concept drift detection and model selection with simulated recurrence and ensembles of statistical detectors", Journal of Universal Computer Science, 2013, (19)4: pp.462-483.
25. W. Nick Street, and Yong Seog Kim, "A Streaming Ensemble Algorithm (SEA) for Large Scale Classification", Proceedings of the seventh ACM SIGKDD, 2001, pp. 377-382.
26. X. Wei, Y. Li, D. Liu, and L. Zhan, Mahalanobis, "Support Vector Machines Made Fast and Robust -

New Advances in Machine Learning", InTech, ISBN: 978-953-307-034-6, China, 2010, pp.227-250.

27. Yi Yang, and Guojun Mao, "A Self-Adaptive Sliding Window Technique for Mining Data Streams", Proceedings of ICEC'13, 2013, pp.689-697.

Dr.S.Jayanthi was born in the year 1981. She received her Bachelor degree in Computer Science from Bharathidasan University, India in 2002, and Master degrees in Computer Applications, and Computer Science and Engineering from Bharathidasan University, in 2005, and from Anna University of Technology, India in 2009, respectively. She obtained her Ph.D from Karpagam University, Coimbatore, India. She has 8 years of teaching experience. Her area of research includes Data Mining, Neural Networks and Big Data analysis. So far she has published 26 papers in various national, International conferences and journals.