ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 6, June 2013

Learning Technique Defined Using Concept Drift in Mining System

Pinaki Barman[1], Mamatha. A[2]

M.Tech Student, CSE, East West Institute of Technology, Bangalore, India[1]

Assistant Professor, CSE, East West Institute of Technology, Bangalore, India[2]

Abstract: Machine learning approach has got major importance when distribution of data is unknown. Classification of data from the data set arises some problem when distribution of data is unknown. Characterization of raw data relates to whether the data can take on only_discrete values or whether the data is continuous. In real world application data drawn from non-stationary distribution, arise the problem of "concept drift" or "non-stationary learning". Drifting of dataset is often associated with online learning scenario. There are several approaches to track the drift from the dataset; detection of drift has got major research attention. One of the problems of filtering is that it cannot detect concepts change or drift happens as time goes accurately. To deal with the concept drift this paper shows some results of different kind of approach for various kinds of datasets. Detection of drift works for two different levels; warning, and alarm level.

Keywords: Machine Learning, concept drift, Diversity, classification, Bagging, Boosting, Poission Distribution, Ensemble.

I. INTRODUCTION

Classification learning from a static dataset can be done easily. So, it is assumed that the dataset contain all necessary information to learn the relevant concepts. The model which is working in real world scenarios, e.g., intrusion detection, spam detection, fraud detection, loan recommendation, climate data analysis makes some prediction on previous data to detect the upcoming changes [1]**.** All training data often received over time in streams of instances or batches. Arrival of data takes different ways either incrementally or in batches. Learning of model using all the information predicts new instances arriving at time step t+ 1.

Online Learning processes the training example in small chunks where as incremental learning process in large data [2]. A learning algorithm is incremental when it produces a sequence of depends on the training data and a limited number of previous hypotheses. A classifier can be updated incrementally from newly available data and simultaneously maintaining the performance of the classifier on old data. Stability of classifier evaluated when it is learning through the changing dataset and adaptive to the new concept. Concept [3] change causes classification problem, as received emails changes as time goes by. This paper shows classifier's accuracy in classifying different kind of datasets. It shows some results where a single best classifier has greater stability than an ensemble of classifier.

the rest of paper is organized as follows: Section 2 Related work, Section 3 Learning challenges from data stream , Section 4 Concept drift, Section 5 Windowing technique , Section 6 Drift detector, Section 7 Ensemble technique

II. RELATED WORK

Fixed size of window on training data in machine learning is not efficient enough than adaptive size of window. Another approach of weighting examples has been used for information filtering in the incremental approach; incremental learning approaches give emphasis to the plasticity of the classifier to learn a new chunk of data. Determining the chunk size is very difficult to learn the new concept because a too small chunk size will not provide enough data for a new classifier to be accurate, whereas a too large chunk size may contain data belonging to different concepts. Frequent changes in the dataset arise great problem in making the adaptation to new concepts..

In offline mode, diversity among base learners is an issue that has been receiving lots of attention in the ensemble learning literature. The success of ensemble learning algorithms is believed to depend both on the accuracy and on the diversity among the base learners (Dietterich; 1997) and some empirical studies revealed that there is a positive correlation between accuracy of the ensemble and diversity among its members (Dietterich; 2000; Kuncheva and Whitaker; 2003). Breiman (2001) also shows that random forests with lower generalization error have lower correlation among base learners and higher base learners' strength. Besides, he derives an upper bound for the generalization error of random forests which depends on both correlation and strength of the base learners. In regression tasks, the bias-variance-covariance decomposition (Ueda and Nakano; 1996) can provide a solid quantification of diversity for linearly weighted ensembles. The decomposition shows that the mean squared error of an ensemble depends critically on the amount of correlation between networks, quantified in the

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 6, June 2013

covariance term and that; ideally, we would like to decrease the covariance at the same time as being careful not to increase the bias and the variance terms.

III. LEARNING CHALLENGES FROM DATA STREAM

Traditional data mining generates dataset from a single, static source but difficulty in learning the data arises, when source of data is different. From the streaming data the function which generates instances at time step t need not be the same function as the one that which generates instances at time step(t+1). This variation in the underlying function is known as concept drift. The major assumption with concept drift **[4]** is that the generating function of the new data is unknown to the learner, and hence the concept drift is unpredictable. If the generating function for the drifting concepts was known, one could merely learn an appropriate classifier for each relevant concept, and apply the correct classifier for all new data .In the absence of such knowledge, then, we must design an ensemble of classifier that can handle such changes in concepts over time.

Another challenge arises when it is assumed that each class in the dataset is will remain, equivalent. While class in traditional data mining problems remains constant, such an assumption is particularly impractical in streaming data applications, where the class distributions can become severely imbalanced. Learning is a sequence of trial and error method. In each trial, the algorithm receives an instance from some fixed domain and is to produce a binary prediction. At the end of the trial, the algorithm receives a binary label, which can be viewed as the correct prediction for the instance. Several real-world applications operate in this sort of scenario, such as spam detection, prediction of conditional branches in microprocessors, information filtering, face recognition, etc. The system might be required to make predictions on instances belonging to both the old and new concepts. Work like survey and detection of various diseases based approaches are shown in [13], [14], [15].

IV.CONCEPT DRIFT

Learning from data streams, we assume that at time step t the learning algorithm H is presented with a set of labeled instances$\{L_0 , . . . ,L_t \}$, where L_i is a p-dimensional feature vector and each in-stance has a corresponding class label y_i. Given an unlabeled instance L_{t+1}, the learning algorithm provides a (potentially probabilistic) class label for L_{t+1}. Once the label is predicted, we assume that the true label $y_{t+1}$and a new testing instance L_{t+2} become available for testing. Furthermore, we call the hidden function f generating the instance at time t as f_t.

Concept drift [5] occurs when the underlying data stream generation function (f) changes over time. There are multiple ways in which this change can occur. Consider classifying L_{t+1}: in order to optimally classify L_{t+1}, we need to know two pieces of information. First, the prior probability of observing each class,$p(c_i)$, and second, the conditional probability of observing L_{t+1} given each class. $p(L_{t+1}|c_i)$. Bayes' theorem then allows us to compute the probability that $L_{t+1}$is an instance of class c_i as:

$$p(c_i|L_{t+1})= p(c_i)\, p(L_{t+1}|c_i)/\quad(L_{t+1}) \qquad (1)$$

Where $p(L_{t+1})$ is the probability of observing L_{t+1}. Note, however, that $p(L_{t+1})$ is constant for all classes ci, and can thus be ignored. Concept drift can then occur with respect to any of the three major variables in Bayes' theorem:

1. $p(c_i)$ may change (class priors).
2. $p(L_{t+1}|c)$ may change (the distributions of the classes).
3. $p(c|L_{t+1})$ may change (the posterior distributions of class membership).

The prior probability of the instances increases after concept drift, the change in $p(c_i)$; the first type of concept drift. Such concept drift can be problematic, as the change in class priors can cause well calibrated classifiers to become miscalibrated. The second type of concept is a change in $p(L_{t+1}|c)$.

Finally the posterior probability of an instance belonging to a particular class changes after concept drift, this uncertainty, due to a change in $p(c|L_{t+1})$, is the most severe form of concept drift, because it directly affects the performance of a classifier, as the distribution of the features, with respect to the class, has changed.

V. WINDOWING TECHNIQUE

The most popular approach to dealing with time changing data involves the use of sliding windows [6]. The procedure of using sliding window for mining data stream is suggested because it has the property of anytime learning and able to provide the best answer after each example. The basic windowing algorithm follows straightforward approach. Each example replaces the data in the window and later the classifier is learned by that window. The key part of this sliding window technique is learning the classifier through forgetting process. In the general approach of sliding window technique the size of sliding windows has fixed size and includes only the most recent examples from the data stream. If someone chooses a small window size the classifier will react quickly to changes, but may loose on accuracy in periods of stability, choosing a large size will increase the accuracy in periods of stability but fails to adapt the sudden changes.

Consider an example, our objective is how to increase the distance between two consecutive error with classifier learning method. Because, the more the leaner will learn, it will have correct prediction of drift in the distribution of data.

0	0	0	0	1	0	0	0	1	0
1	2	3	4	5	6	7	8	9	10

Here, when classifier will detect drift, at the time it will get drift '1' after consecutive four '0', same again will get drift after consecutive three '0'. So, if we use a classifier which is learned with only '0' digit not with '1' ,that

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 6, June 2013

classifier will not be able to classify the whole window properly. Instead of that, if had been classified with two different types of classifier one is trained with '0' and another one is trained with '1' then whole window would be properly classified. It has always been found that ensemble of classifier is better than the single base classifier

Then the choice of size of sliding window looks upon the way of learning the classifier. It is always better to use the dynamic ways of modeling the size of window in the forgetting process.

Algorithm 1: The basic windowing algorithm
Input: S: a data stream of examples
W: window of examples
Output: C: a classifier built on the data in window W
1: initialize window W;
2: for all examples $x_i \in S$ do
3: $W \leftarrow W \cup \{x_i\}$;
4: if necessary remove outdated examples from W;
5: rebuild/update C using W;

VI. DRIFT DETECTOR

There is several learning algorithm to detect the changes which are efficient depending on the way of learning approach. Both the learning approaches online and offline have to be adaptive to the change from the evolving data stream. Detector of drift from the database makes an alarm to the base learner that its classifier should be updated. Statistical test provides enough methods that verify the running error or class distribution remain constant over time.

I.DDM

Gama [7] based their Drift Detection Method (DDM) on the fact, that an online classifier predicts the decision class of an example. That prediction can be either true or false, thus for a set of examples the error is a random variable from Bernoulli trials. That is why the authors model the number of classification errors with a Binomial distribution. Let us denote p_i as the probability of a false prediction and s_i as its standard deviation calculated as given by

$$S_i = \sqrt{pi(1-pi))/i} \qquad (2)$$

The binomial distribution gives the discrete probability distribution of $P_P(N|n)$, there n successes out of N Bernoulli trial is closely approximated by a Normal distribution with the same mean and variance. The *binomial distribution* has to maintain some properties for the variable *X*. Properties can be listed like this :(1) The number of observations n is fixed.(2) Each observation is independent.(3)Each observation represents one of two outcomes ("success" or "failure").(4)The probability of "success" p is the same for each outcome. For each example, error rate from the data stream be tracked updating two registers: p_{min} and s_{min}. These values are used to calculate a warning level and alarm level. If an example of window reaches to warning level then that window is remembered in a separate window. If afterwards the error rate is lesser than the warning threshold, then it is assumed as false alarm. However, if the alarm level is reached, the previously taught base learner is dropped and a new one is created from the examples stored in the separate "warning" window.

$$p_i + s_i \geq p_{min} + 2 \cdot s_{min} \qquad (3)$$
$$p_i + s_i \geq p_{min} + 3 \cdot s_{min} \qquad (4)$$

The authors proposed $\alpha = 2$ and $\beta = 3$, giving approximately 95% confidence of warning and 99% confidense of drift. DDM works best on data streams with sudden drift. When no changes are detected, DDM works like a lossless learner constantly enlarging the window size which can lead to the problem of memory limitation.

II.EDDM

The authors use the same warning alarm mechanism that was proposed by Gama, but instead of using the classifier's error rate, they propose the distance error rate. They denote p_i' as the average distance between two consecutive errors and s_i' as its standard deviation. Using these values the new warning and alarm conditions are given by Equation

$$\frac{pi' + 2 \cdot si'}{pmax' + 2.smax'} < \alpha \qquad (5)$$

$$\frac{pi' + 3 \cdot si'}{pmax' + 3.smax'} < \beta \qquad (6)$$

EDDM works better than DDM for slow gradual drift, but is more sensitive to noise.

VII. ENSEMBLE TECHNIQUE

In the recent years many ensemble methods is popular in the data mining community due in part to their empirical effectiveness. Specifically tailored classifier has shown high stability towards mining from data streams. In addition, classifier has to be able to cope with concept drift where scalability is the most important issue. For very large datasets, the classifiers like decision trees is not very efficient, in that case Bayesian classifier is very useful. Learners are trained with some slightly different datasets in ensemble methods to avoid over fitting, ensure that the ensemble is diverse. Divers class of learner in ensemble method are not all similarly biased when making predictions. Some examples of traditional ensemble methods are bagging [9], AdaBoost [10], random forest [11]. In making prediction of drift from database, ensemble based approaches have shown greater accuracy. Advantage of ensemble technique is their ability to deal with reoccurring concepts from streaming of datasets.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 6, June 2013

Since ensemble of classifier is learned from past data, such models can be reused to classify new instances. In SEA algorithm [12] breaks the stream into a series of consecutive, non-overlapping windows. For each new window, a new model is learned on all of the instances from that window. If the current size of ensemble is not full, the new model is added to the ensemble. Otherwise, the model is tested against all other models currently in the ensemble, and the "worst" one is pruned. In order to determine which classifier to prune, Street and Kim recommend a classifier replacement strategy based on instances ,but it is "nearly undecidable " because if the particular instance has no significant effect on the classifier. Voting strategy another kind of approach where ensemble members are spilited up based on their class labeling. Voting result based on the instance has higher impact on the retention (or removal) of the classifier. This approach perform well on the instances which are not easy to be classified, while simultaneously ignoring the classifier's performance on "impossible" instances making the ensemble more robust to noise. The original online bagging (Algorithm 2) is based on the fact that, when the number of training examples tends to 1 in offline bagging, each base learner h_m contains K copies of each original training example, where the distribution of K tends to a Poission(1) distribution. So, in online bagging, whenever a training example is available, it is presented K times for each base learner h_m, where K is drawn from a Poisson (1) distribution. The classification is done by un-weighted majority vote, as in offline bagging.

Algorithm 2.Online Bagging

Input: ensemble h; ensemble size M; training example d; online learning algorithm for the ensemble members

1. For m 1 to M do
2. K Poission(1)
3. While K
4. $h_m \leftarrow$ OnlineBaseLearningAlg(h_m,d)
5. K $\leftarrow$ K – 1
6. End while
7. End for

Output: updated ensemble h

.

VIII. EXPERIMENTAL OBJECTIVE, DESIGN and MEASURES

The objective of the experiments with DWM and EDDM is to assist its analysis and to check the accuracy on dataset of Cancer Survival. We also aim at identifying for which types of drift works better and why it behaves in that way. There are many ensemble approach, in order to do so, we analyse measures the number of change detected and number of warning detected and prequential accuracy. In some cases, the false positive and negative rate is also analysed with no drift handling abilities, EDDM, and DWM. The prequential accuracy is calculated based on the predictions given to the current training example before the example is used for updating any component of the system. It is important to observe that updating the model is not only important on the current training example by the base learners, but also to the changes in weights associated with the base learners. The prequential accuracy is compared both visually, considering the graphs of the average prequential accuracy and standard deviation throughout the learning, and using T student statistical tests.

This paper shows results with some snap shots, the changes in the datasets. Classes of attributes can be shown from figure (1). DWM model dynamically add or remove the weight from classifier. In an ensemble approach, the classifier which one incorrectly classifies data will get higher weights to be processed for more time

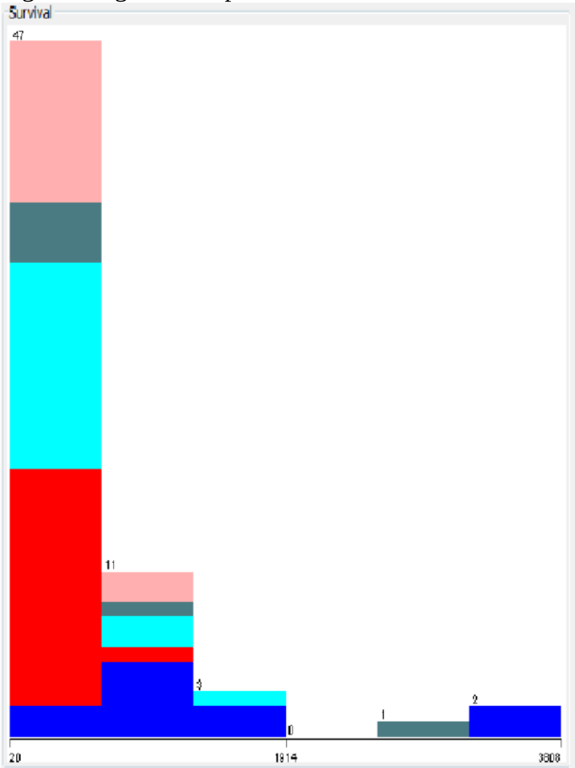

Figure 1:Cluster diagram

It has been found that in classifying the Cancer survival data by a single classifier like naïve bayes classifier may classify the data with higher accuracy.

Accuracy is calculated; how many instances have been classified properly out of total instances. So, utilization of Classifier has great impact on classifying the dataset.If any instances is correctly classified means prediction value and actual value is same then we it can reduce the mean absolute error.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 6, June 2013

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          25               39.0625 %
Incorrectly Classified Instances        39               60.9375 %
Kappa statistic                          0.1832
Mean absolute error                      0.2879
Root mean squared error                  0.3988
Relative absolute error                 91.901  %
Root relative squared error            100.6888 %
Total Number of Instances               64

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.455     0.057      0.625     0.455     0.526      0.804    Breast
                 0.882     0.574      0.357     0.882     0.508      0.664    Bronchus
                 0.294     0.191      0.357     0.294     0.323      0.563    Colon
                 0         0          0         0         0          0.273    Ovary
                 0         0          0         0         0          0.425    Stomach
Weighted Avg.    0.391     0.213      0.297     0.391     0.311      0.576

=== Confusion Matrix ===

  a  b  c  d  e   <-- classified as
  5  2  4  0  0 |  a = Breast
  0 15  2  0  0 |  b = Bronchus
  1 11  5  0  0 |  c = Colon
  2  3  1  0  0 |  d = Ovary
  0 11  2  0  0 |  e = Stomach
```

Figure 2: Naïve Bayes approach

In naïve Bayes approach out of 64 instances 25 are correctly classified and 39 are incorrectly classified. To have consistent accuracy of classifier our goal is to reduce the MSE error. Another kind of approach of ensemble of classifier in bagging method, increasing the number of iteration we can improve the accuracy of classifier. Output result shows in figure 3.

Accuracy is .39. Here 5 instances is correctly classified as Breast cancer.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          26               40.625  %
Incorrectly Classified Instances        38               59.375  %
Kappa statistic                          0.2073
Mean absolute error                      0.2859
Root mean squared error                  0.3913
Relative absolute error                 91.2725 %
Root relative squared error             98.8156 %
Total Number of Instances               64

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.545     0.075      0.6       0.545     0.571      0.823    Breast
                 0.824     0.511      0.368     0.824     0.509      0.698    Bronchus
                 0.353     0.213      0.375     0.353     0.364      0.606    Colon
                 0         0          0         0         0          0.239    Ovary
                 0         0          0         0         0          0.505    Stomach
Weighted Avg.    0.406     0.205      0.301     0.406     0.33       0.613

=== Confusion Matrix ===

  a  b  c  d  e   <-- classified as
  6  2  3  0  0 |  a = Breast
  0 14  3  0  0 |  b = Bronchus
  1 10  6  0  0 |  c = Colon
  2  2  2  0  0 |  d = Ovary
  1 10  2  0  0 |  e = Stomach
```

Figure 3: Bagging output

In bagging method accuracy is .40. Here 6 instances is correctly classified as Breast cancer.

Here number of correctly classifying instances has been increased and also MSE error has been reduced. Voting method takes vote from each member to predict the class of instance. Model with lower predicted value will be rejected from the voting model at first. Here voting method takes result from naïve bayes classifier and J48 algorithm .Voting method works on the concept of which are the classifier has correctly classified the data, if the number of classifier is greater than the classifier has incorrectly classified the data then voting result will support the greater in number.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 6, June 2013

```
=== Summary ===

Correctly Classified Instances          22               34.375  %
Incorrectly Classified Instances        42               65.625  %
Kappa statistic                          0.1379
Mean absolute error                      0.2893
Root mean squared error                  0.4089
Relative absolute error                 92.3296 %
Root relative squared error            103.2492 %
Total Number of Instances               64

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.545     0.132     0.462     0.545     0.5        0.81     Breast
                 0.647     0.34      0.407     0.647     0.5        0.67     Bronchus
                 0.176     0.277     0.188     0.176     0.182      0.496    Colon
                 0         0         0         0         0          0.328    Ovary
                 0.154     0.118     0.25      0.154     0.19       0.457    Stomach
Weighted Avg.    0.344     0.21      0.288     0.344     0.306      0.572

=== Confusion Matrix ===

  a  b  c  d  e   <-- classified as
  6  1  3  0  1 |  a = Breast
  0 11  4  0  2 |  b = Bronchus
  4  7  3  0  3 |  c = Colon
  2  2  2  0  0 |  d = Ovary
  1  6  4  0  2 |  e = Stomach
```

Figure 4: voting output

On this relevant dataset, voting method does not have that higher accuracy than bagging algorithm. AdaBoost method updates each model for all the samples and for each iteration output of previous model taken as input to next model. Output result Shows in figure 5.

```
Correctly Classified Instances          23               35.9375 %
Incorrectly Classified Instances        41               64.0625 %
Kappa statistic                          0.1447
Mean absolute error                      0.2894
Root mean squared error                  0.397
Relative absolute error                 92.3749 %
Root relative squared error            100.2524 %
Total Number of Instances               64

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.455     0.094     0.5       0.455     0.476      0.833    Breast
                 0.824     0.532     0.359     0.824     0.5        0.688    Bronchus
                 0.235     0.234     0.267     0.235     0.25       0.588    Colon
                 0         0         0         0         0          0.26     Ovary
                 0         0         0         0         0          0.418    Stomach
Weighted Avg.    0.359     0.22      0.252     0.359     0.281      0.591

=== Confusion Matrix ===

  a  b  c  d  e   <-- classified as
  5  2  4  0  0 |  a = Breast
  0 14  3  0  0 |  b = Bronchus
  2 11  4  0  0 |  c = Colon
  2  2  2  0  0 |  d = Ovary
  1 10  2  0  0 |  e = Stomach
```

Figure 5: Boosting output

Applying DDM method on Cancer Survival dataset we have this kind of projected result of accuracy

```
learning evaluation instances     =64.0
evaluation time (cpu seconds)     =0.0468003
model cost (RAM-Hours)            =-1.2107271080215773E-14
classified instances              =64.0
classifications correct           =37.5
Kappa Statistic (percent)         =19.471531928279333
model training instances          =29.0
model serialized size (bytes)     =-1.0
change detected                   =1.0
Warning detected                  =12.0
```

Figure 6: DDM method

And applying EDDM method on the same dataset we are having some different kind of result. In this method classification accuracy is higher than general DDM approach.

```
learning evaluation instances =64.0
evaluation time (cpu seconds) =0.0780005
model cost (RAM-Hours)        =-2.0178785133692953E-14
classified instances          =64.0
classifications correct       =51.5625
Kappa Statistic (percent)     =37.78613985575415
model training instances      =12.0
model serialized size (bytes) =-1.0
change detected               =1.0
Warning detected              =0.0
```

Figure 7: EDDM method

ACKNOWLEDGMENT

Special thanks to East West Institute of Technology for helping us in our work and supporting the research. Visvesvaraya Technological research centre are also helping us in this work

REFERENCES

[1] T. Ryan Hoens · Robi Polikar · Nitesh V. Chawla "Learning from Streaming Data with Concept Drift and Imbalance: An Overview," IEEE Intelligent Systems, vol. 25,no. 4, pp. 1-12,.
[2] Leandro L. Minku, "The Impact of Diversity on Online Ensemble Learning in the Presence of Concept Drift," IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING,pp731-742, 2010.
[3] Haixun Wang Wei Fan Philip S. Yu 1,Jiawei Han, "Mining Concept-Drifting Data Streams using Ensemble Classifiers" Univ. of Illinois, Urbana, IL 61801,pp, 2002.
[4] L. Tang, S. Rajan, and V. K. Narayanan, "A General Framework for Mining Concept-Drifting Data Streams with Skewed Distributions,," Proc. SIAM Int'l Conf. Data Mining (ICDM), 2007.
[5] J.Z. Kolter and M.A. Maloof,, "Using Additive Expert Ensemblesto Cope with Concept Drift," Proc. Int'l Conf. Machine Learning (ICML '05), pp. 449-456, 2005*FLEXChip Signal Processor (MC68175/D)*, Motorola, 1996.

[6] L.I. Kuncheva and C.J. Whitaker,, "Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy" Machine Learning, vol. 51, pp. 181-207, 2003.
[7] Ricard Gavald`, and Rafael Morales-Bueno, "Early Drift Detection Method," in Departmental de Languages' y Ciencias de la Computaci´on. E.T.S. Ingenier´ıa Inform´atica. Universidad de M´alaga, Spain. 2005
[8] W. Fan, "Streamminer: A Classifier Ensemble-Based Engine to Mine Concept-Drifting Data Streams" Proc. 30th Int'l Conf. VeryLarge Data Bases, pp. 1257-1260, 2004.
[9] Mine Concept-Drifting Data Streams M. Scholz and R. Klinkenberg, "An Ensemble Classifier for Drifting Concepts," Proc. Second Int'l Workshop Knowledge Discovery from Data Streams, pp. 53-64, 2005
[10] F. Chu and C. Zaniolo "Fast and Light Boosting for Adaptive Mining of Data Streams" Proc. Pacific-Asia Conf. Knowledge Discovery and Data Mining (PAKDD '04), pp. 282-292, 2004.
[11] H. Abdulsalam, D.B. Skillicorn, and P. Martin, "Streaming Random Forests", Proc. Int'l Database Eng. and Applications Symp.(IDEAS), pp. 225-232, 2007
[12] W. Street and Y. Kim "A Streaming Ensemble Algorithm (SEA) for Large-Scale Classification," P roc. ACM Conf. Knowledge Discovery and Data Mining (KDD), pp. 377-382, 2001.
[13] Hemant, Palivela; Prof. Yogish H K, Vijaykumar S, Kalpana Patil;, " A Study of Mining Algorithms for Finding Accurate Results and Marking Irregularities in Software Fault Prediction," IEEE Sponsored International Conference On INFORMATION COMMUNICATION & EMBEDDED SYSTEMS "ICICES 2013" , 21-23 February 2013, ISBN : 978-93-82-880-03-5.
[14] Hemant, Palivela; Prof. Yogish H K, Vijaykumar S, Kalpana Patil; , " A Survey on mining algorithms for Breast cancer related data," IEEE Sponsored International Conference On INFORMATION COMMUNICATION & EMBEDDED SYSTEMS "ICICES 2013" , 21-23 February 2013, ISBN : 978-93-82-880-03-5.
[15] Hemant Palivela, Pushpavathi Thotadara ,Computing Communication & Networking Technologies (ICCCNT), 2012 Third International Conference on computing communication and network technologies, 26th -27th July 2012.