IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-6, Issue-3 E-ISSN: 2347-2693

Significance of learning methods for mining of real time data streams

E.Padmalatha, S.Sailekya

Dept. of CSE, CBIT, Bvrith, India

Available online at: www.ijcseonline.org

Received: 16/Feb//2018, Revised: 21/Feb2018, Accepted: 15/Mar/2018, Published: 30/Mar/2018

Abstract- Stream Data is now more than ever highly distributed, loosely structured, increasingly large in volume and changing over time. Broadly speaking, firstly the volume of data increasing exponentially each year and secondly the speed at which the new data is being generated of distinct concept and changes over time. Stream Data is generated by number of sources. Data streaming applications are typically dealing with large amounts of data over an extended period of time. However, in most cases the user is only interested in recent data instead of the whole data set. Furthermore, stream data tends to express features of a concept drift, i.e. the data is evolving over time. This would cause algorithms which consider the whole data set with the same importance to produce distorted results. In such cases the majority of processed data would not be valid anymore. Sometimes the nature of a data stream itself requires giving up a certain amount of precision because its high volume couldn't be processed otherwise and one would end up with no information at all. If the data distribution is stable, mining a data stream is largely the same as mining a large data set, since statistically it is easily to mine a sufficient sample. The expectations of mining data streams are finding and understanding changes, maintaining an updated model. For evolving data, two classes of problems are of particular interest: model maintenance and change detection. The goal of model maintenance is to maintain a data mining model under inserts and deletes of blocks of data. In this model, older data is available if necessary. Change detection is related to quantify the difference between two sets of data and determine when the change has statistical significance. Data streams can be seen as stochastic processes in which events occur continuously and independently from each another [1]. Querying data streams is quite different from querying in the conventional relational model. A key idea is that operating on the data stream model does not preclude the use of data in conventional stored relation, data might be transient.

In this paper proposed methods are addressing Classification of balanced and unbalanced data streams by considering concept drift and data skewness. The classification accuracy depends on the selection of learning model. In data streams at the time of classification ,concept drift plays the vital role .Comparing to traditional classification data stream classification needs more accurate methods .Because traditional methods always follows the training model which may not predict the novel classes. In data streams by considering the concept drift with unsupervised learning model can predict the novel class. In the proposed methodology classification of data streams are addressed by ensemble methods with supervised learning, unsupervised learning for novel class detection to increases the accuracy of the system. A scalable and adaptable online genetic algorithm is proposed to mine classification rules for the largest data streams with concept drifts. The data skewness is addressed by considering the data level, the algorithmic level to favor the positive class.

Keywords- Data Mining

1. Introduction

1.1 Data Streams

A data stream is an ordered sequence of instances that arrive at a rate that does not permit to permanently store them in memory. Data streams are potentially unbounded in size making them impossible to process by most data mining approaches. The main characteristics of the data stream model imply the following constraints [2], it is impossible to store all the data from the data stream. Only small summaries of data streams can be computed and stored, and the rest of the information is thrown away. The arrival speed of data stream tuples forces each particular element to be processed essentially in real time, and then discarded. The distribution generating the items can change over time. Thus, data from the past may become irrelevant or even harmful for the current summary.

1.2 Learning from Data Streams Hulten and Domingos (2001) identify desirable properties of learning systems for efficient mining continuous, high-volume, open-ended data streams:

- Require small constant time per data example.
- Use fix amount of main memory, irrespective to the total number of examples.
- Built a decision model using a single scan over the training data.
- Generating a anytime model independent from the order of the examples.
- Ability to deal with concept drift.
- For stationary data, ability to produce decision models that are nearly identical to the ones it can be obtained by using a batch learner.

From these desiderata, three dimensions can be identified which influence the learning process: space- the available memory is fixed, learning time ,process incoming examples at the rate they arrive, and generalization power how effective the model is at capturing the true underlying concept.

1.3 Concept Drift

An additional problem of the holdout method comes from the non-stationary properties of data streams. Non-stationarity or concept drift means that the concept about which data is obtained may shift from time to time, each time after some minimum permanence.

1.4 Data Skewness

As data streams become increasingly ubiquitous and prolific, the importance of solving their unique and interrelated challenges grows. Streaming data is pervasive in a multitude of data mining applications. One fundamental problem in the task of mining streaming data is distributional drift over time. Streams may also exhibit high and varying degrees of class imbalance, which can further complicate the task in decision making. A dataset for modelling is perfectly balanced when the percentage of occurrence of each class is 100/n, where n is the number of classes. If one or more classes differ significantly from the others, this dataset is called skewed or unbalanced [3].

2. Literature survey

2.1 Data stream cycle

In the data stream model, data arrive at high speed, and an algorithm must process them under very strict constraints of space and time. A data stream environment has different requirements from the traditional batch learning setting. The most significant are the following:

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

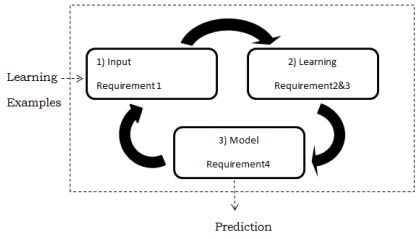


Fig1. Data stream life cycle

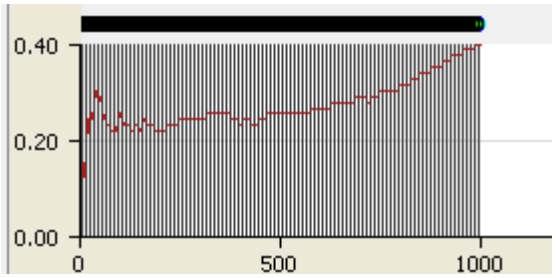
Figure 1 illustrates the typical use of a data stream classification algorithm, and how the requirements fit in a repeating cycle:
1. The algorithm passes the next available example from the stream (Requirement 1).
2. The algorithm processes the example, updating its data structures. It does so without exceeding the memory bounds set on it (requirement 2), and as quickly as possible (Requirement 3).
3. The algorithm is ready to accept the next example. On request it is able to predict the class of unseen examples (Requirement 4).

2.2 Learning with concept drift

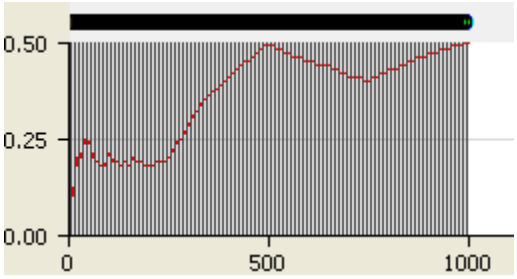
In the real world, concepts are often not stable but change with time. Typical examples of this are weather prediction rules and customers' preferences. The underlying data distribution may change as well. Often these changes make the model built on old data inconsistent with the new data, and regular updating of the model is necessary. This problem is known as *concept drift*. Concept Drift between time point t_0 and time point t_1 defined as $\exists X: pt = \pi r^2$

Where p_{t0} denotes the joint distribution at time t_0 between the set of input variables X and the target variable y. changes in data can be characterized as changes in the components of the relation [4, 5], the prior probabilities of classes p(y) the class conditional probabilities p(X | y) may change ,and as the result ,the posterior probabilities of classes p(y | X) affects the prediction.

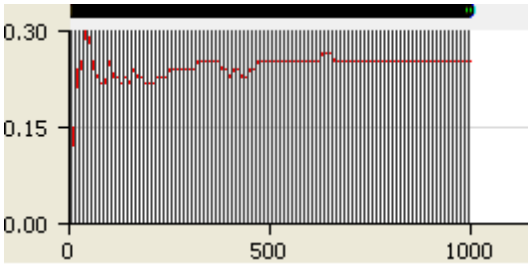
A change in the instances selected for learning the decision model could also allow a system to acclimate to a concept drift. Most of the approaches studied in this thesis are referred to as incremental learning approaches. This means they make decisions about one or more current objects or situations based on past observations. The current model is evaluated by how it performs on the latest observation. Incremental learning and the term "concept drift" were introduced by Schlimmer and Granger [6]. Three types of concept drifts pay major role in study of data streams which are shown in fig2a, b, and c.



(a)



(b)



©

Fig2 a) Gradual change b) Abrupt change , c) No change

2.3 Issues with Data skew

A Dataset is unbalanced when the class of interest (minority class) is much smaller or rarer than normal behavior (majority class). Classification algorithms in general suffer when the data is skewed towards one class. In this poster we present a comparison of existing methods for dealing with unbalanced data.

Unbalanced problem

- The cost of missing a minority class is typically much higher than missing a majority class.
- Most learning systems are not prepared to cope up with large difference between the number of cases belonging to each Classification algorithm underperform when data is unbalanced[7].

The unbalance problem is typical of many applications such as fraud detection, medical diagnosis, text classification, oil spills detection etc.

3. Classification accuracy with ensemble agent process

3.1 Ensemble learning for classification

Ensemble learning is a very important and popular branch of machine learning that applies the philosophy of '4 eyes see more than 2' in Fig3; it learns a model with a range of learners, using specific rules to integrate various learning outcomes. Ultimately, ensemble learning yields more efficient machine learning compared to that possible by a single learner [8]. Classifier ensembles are a common way of boosting classification accuracy. Due to their modularity, they also provide a natural way of adapting to change by modifying ensemble members. Ensemble algorithms are sets of single classifiers whose decisions are aggregated by a voting rule. The combined decision of many single classifiers is usually more accurate than that given by a single component. Studies show that to obtain this accuracy boost, it is necessary to diversify ensemble members from each other. Components can differ from each other by the data they have been trained on, the attributes they use, or the base learner they have been created from. For a new example, class predictions are usually established by member voting.

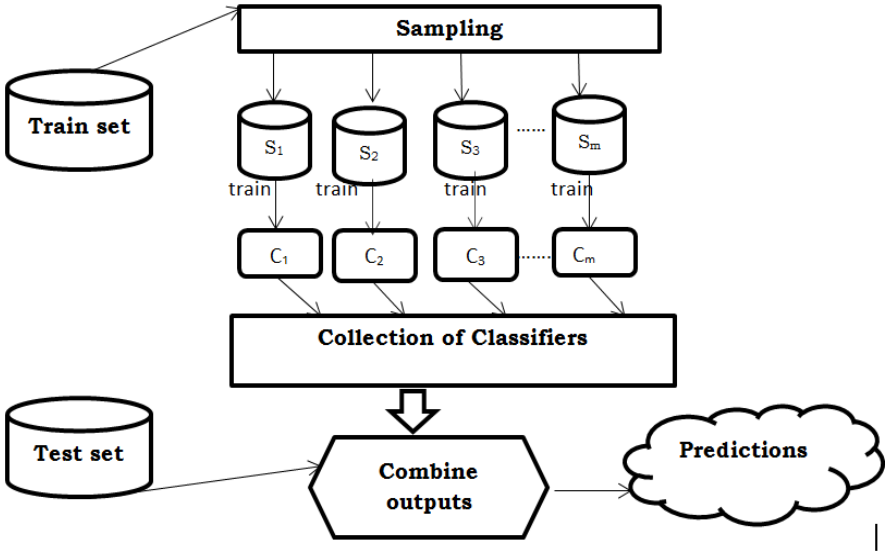


Fig3.Ensemble Classifier

For experiment German credit card dataset [9] is used which is having 1000 instances and 21 attributes. Based on Means squared error (RMSE), and high Kappa statistics two ensemble methods RBF (Random bagging Forest) and JripB(JripBagging) are proposed. The performance of above mentioned methods are pictorially shown with help of the ROC curves based on the confusion matrix. From the experimental results it can conclude that for classification of data streams bagging and Random forest algorithms can be considered. JRIB is not a "Weak "classifier, but is somehow damps the effect of ensemble learning. The ensemble method increased roc value of jripB 0.59 to 0.71, and the roc value of RFB is increased to 0.79 from 0.72.

3.2 Ensemble agents in classification

Heterogeneous multi agent coordination improves accuracy of large and complex data mining task. A working agent is a data mining agent designed to ferret out relevant information to classify the data stream .On one hand, data mining approach can drive the knowledge extraction from huge amount of data; on the other hand, by using a single classifier it may not provide reliable results. Therefore, to improve the classification accuracy rates, ensemble classifiers joint with agents and relied on multiple learning algorithms, can be aid to improve the performance. The main objective of the proposed system is to improve the accuracy of the classification model by using ensemble technique with the data mining agents. Data can be divided into partitions according to the partitions the number of agents will be created and the partitions are processed by all the agents in the systems, the results from the partitions are fused to consider the accuracy .Finally accuracies are compared with respective to those achieved by using classifier.

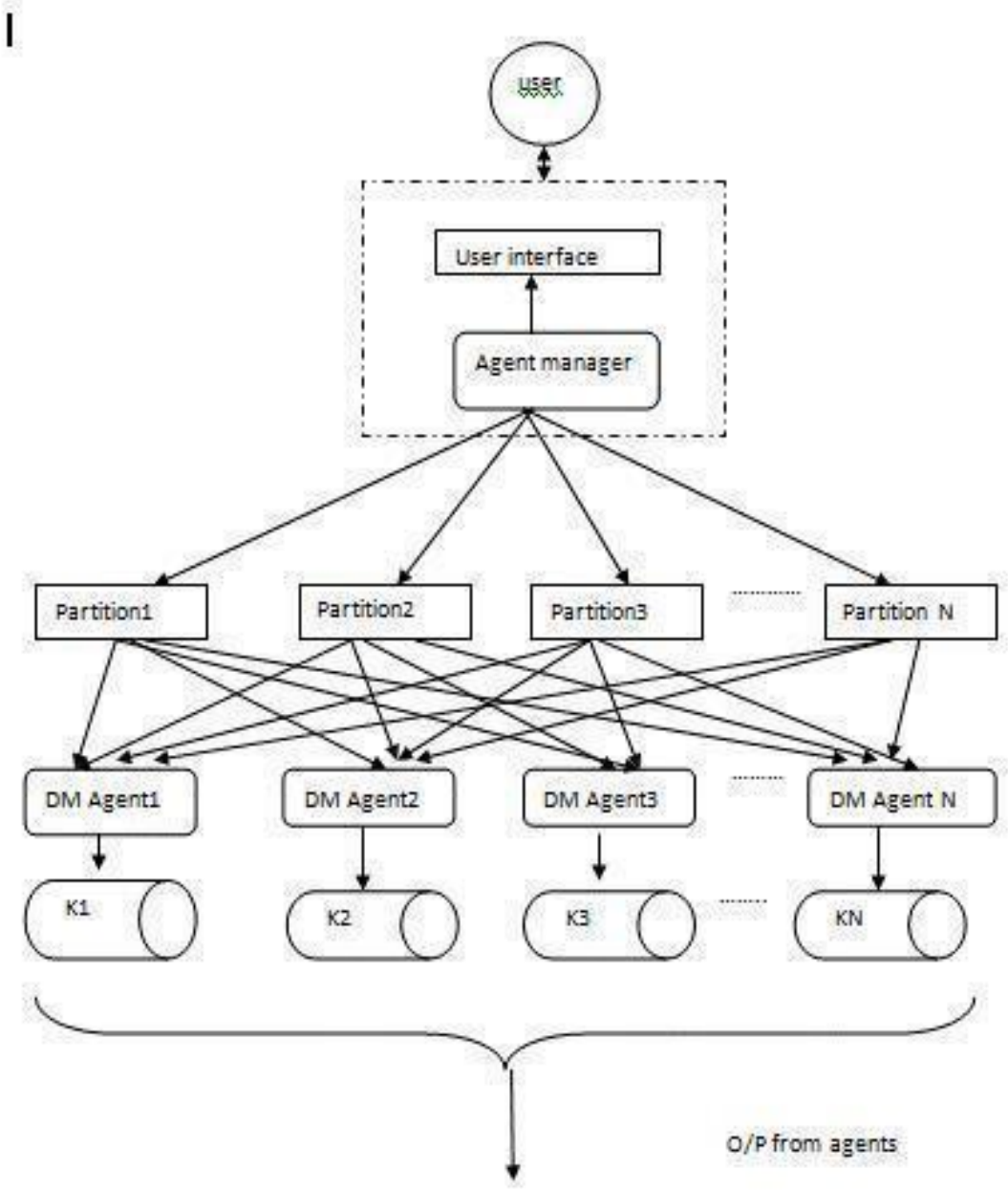


Fig4Classification model with ensemble agents and classifiers

Table.1.Showing the accuracies of different classifiers

Classifiers/Dataset Name	breast – cancer	contact-Lens	Credit-g	Diabetes	Glass
Decision table	73.076	70.83	72.2	73.95	62.14
J48	71.67	83.33	72.2	74.86	66.82
SVM	69.5	70.83	75	77.03	56.07
JRIP	70.09	75	71.7	76	68.6
Ensemble Agent	78.5	88.09	76.2	77.05	82.64

Table 2 Showing confusion matrix of RFB classifier

	True	False
True	642	58
False	175	125

Table 3 Showing confusion matrix of jripb classifier

	True	False
True	645	55
False	205	95

Table 4showing classification details of RFB and JRIPB

	TP rate	FP rate	Precision	Recall	ROC
RFB	0.76	0.43	0.75	0.76	0.79
JRIPB	0.74	0.50	0.72	0.74	0.71

use of ensemble methods is popular in the data mining community due in part to their empirical effectiveness. This effectiveness is derived from combining multiple classifiers trained on similar datasets to provide accurate and robust predictions for future instances. The use of slightly different datasets and/or base learners is important to ensemble methods so as to ensure that the ensemble is sufficiently diverse, as diversity in ensembles directly leads to better, more accurate, ensembles. Based on the experimentation the proposed method Ensemble Agent with heterogeneous classifiers is having significant improvement in the accuracy which is shown in table1. Based on the confusion matrix of RFB and JRIPB as shown in table2 and 3, statistics are calculated using Weka tool .Calculated statistics are shown in the table 3.7; they are compared with Random Forest and JRIP without bagging and based on the statistics of table 4 it can be concluded that values obtained by the proposed system is efficient. Obtained values in table 4 are plotted in a graph that is shown in fig3.3. According to fig3.3 it is proved that RFB is a good ensemble classifier.

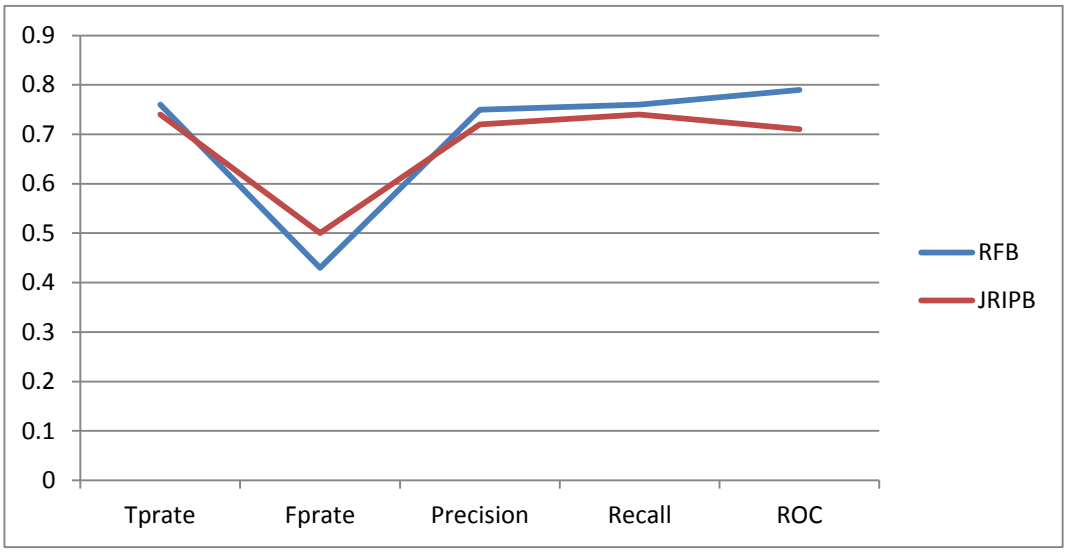


Fig5.Comparison graph of RFB and JRIPB

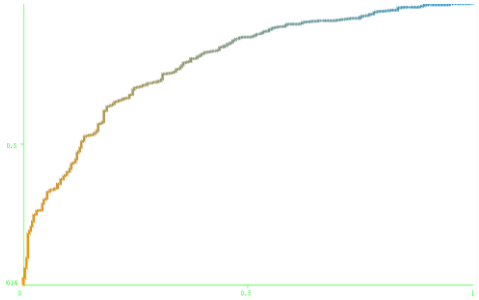


Fig6.RFB showing ROC value=0.79

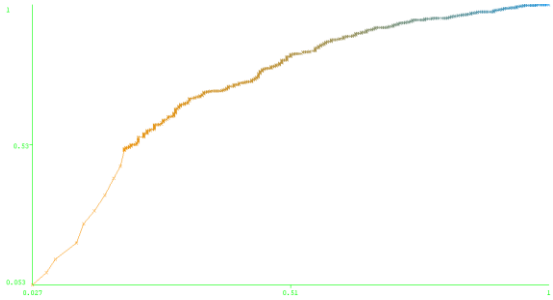


Fig 7 JRIPB ROC value=0.71

By considering TPrate and FPrate of classifier Random Forest with bagging ROC curve was drawn and shown in fig6 .After comparing it with the ROC regions of ROC curve[25] shown in the this ROC curve is nearer to the right corner region which denotes good classifier. Comparing to the fig6 and fig7 it can be shown that when Random Forest is ensemble with bagging it exhibits good accuracy and less false positives.According to fig7 Jrip will be considered as "weak classifier", but when it is ensemble with bagging roc value is increased from 0.59 to 0.71 which is efficient improvement, thus proposed system is proved that ensemble learning will increase the accuracy of the system.

4. Agent based concept drift detection using supervised model

4.1 Agent based CVFDT and CMAC

The proposed Agent basedCVFDT (ACVFDT) and ACMAC neural network based agent are for evolving data streams in distributed network. Here initially different groups of nodes are formed in the distributed network by applying K-means algorithm. The centroid of each group is selected as agent and it is trained using ACVFDT and ACMAC neural network. The agent would check corresponding nodes in the group for concept drift. If the node has concept drift, the ACVFDT is performed to identify the sort of intrusion and if the node has no concept drift, the ACMAC operation is performed for the identification of normal or abnormal data. KDDCUP'99 data set is used for experimentation and the experimentation is processed for three iterations i.e. at three different time intervals, the accuracy that obtained for ACVFDT and CMAC neural network for the three different time intervals is 79.8 for ACVFDT.

4.2The Stream Data Model

This model assumes data arrives at a processing engine at a rate that makes it infeasible to store everything in active storage. One strategy to dealing with streams is to maintain summaries of the streams, sufficient to answer the expected queries about the data. A second approach is to maintain a sliding window of the most recently arrived data.

Rotating Hyperplane It was used as testbed for CVFDT versus VFDT in [10]. A hyperplane in d-dimensional space is the set of points x that satisfy

$$\sum_{i=1}^{d} w_i x_i = w_0 = \sum_{i=1}^{d} w_i$$ ------eq1

Where as in eq1 x_i, is the ith coordinate of x. Examples for which $\sum_{i=1}^{d} w_i x_i \geq w_0$ are labeled positive, and examples for which $\sum_{i=1}^{d} w_i x_i < w_0$ are labeled negative. Hyperplanes are useful for simulating time-changing concepts, because we can change the orientation and position of the hyperplane in a smooth manner by changing the relative size of the weights. We introduce change to this dataset adding drift to each weight attribute $w_i = w_i + d\sigma$, where σ is the probability that the direction of change is reversed and d is the change applied to every example. Proposed system implemented CVFDT and ECVFDT with Rotating hyperplane and obtained accuracies are compared with the VFDT. The state-of-the-art decision tree classification method CVFDT [11] can solve the concept drift problem well, but the efficiency is debased because of its general method of handling instances in CVFDT without considering the types of concept drift.

4.2.1Accidental Concept Drift

The examples with new concepts always have little amount and appears probability in data stream. Traditional CVFDT always uses it to participate in the information gain calculation for best attribute selection. Performance efficiency is reduced and makes the decision selection more complex. To solve this concept drift problem, E-CVFDT algorithm uses a caching mechanism.

4.2.2Gradual Concept Drift

Different evolving concepts flow into the decision tree model to classify. Because of the characteristic of evolving data, the E-CVFDT algorithm will find the best split attribute by computing information gain at a very fast frequency. It's good for obtain a better accuracy of classification, but the performance efficiency needed to reconsider. The complex information gain calculation with evolving data distribution affects the performance efficiency. So E-CVFDT [12] method regroups the data in memory, delays the new concept examples, and lets the original concept evolve in information gain calculation first, it must be useful for improving the performance efficiency.

4.2.3Instantaneously Concept drift

This type of concept drift in data stream is very easy to deal with. Because, after t times, the new concept occurs instantaneously in data stream, at the same time, the old concept disappear, and it will never occurs again. The system maintains only the new ones. The traditional CVFDT algorithms addresses concept drift in this scene quit well, it can achieve good accuracy of classification and comfortable performance efficiency. The E-CVFDT adds a mechanism into traditional CVFDT algorithm, which just only regroups data distribution, and its time complexity is linear. There is no effect for process of creating decision tree of traditional CVFDT, thus, the E-CVFDT algorithm's performance as well as the CVFDT algorithm. For the Implementation of CVFDT and E-CVFDT three different hyperplane datasets are considered. Initially E-CVFDT with higher accuracy values compared to CVFDT algorithm but as we increase the size of the window the accuracy of CVFDT increases and overcomes E-CVFDT accuracy as shown in Figure 5.3 but E-CVFDT produces a constant accuracy thus giving us guarantee of giving a fixed accuracy value independent of window size. Though CVFDT shows greater value of accuracy, it does not work in real time scenarios. It slowly betrays the window concept. As discussed in earlier, one cannot store instances of data stream due to infinite size of data. When the size of the window is increased, the number of instances available in window increases thus it results in classification of similar instances directly without any miss in the tree which eventually reduces the overhead of adding new concept to the tree. This causes drawback of having old concept reside in the tree, causing increase in the storage cost.

Table 5.Showing the results of ACVFDT and E-CVFDT

Hyper plane dataset with different parameters of Drift	ACVFDT	E-CVFDT
Hyper plane 1	69.16	71.45
Hyper plane 4	68.71	71.45
Hyper plane 7	68.98	71.45

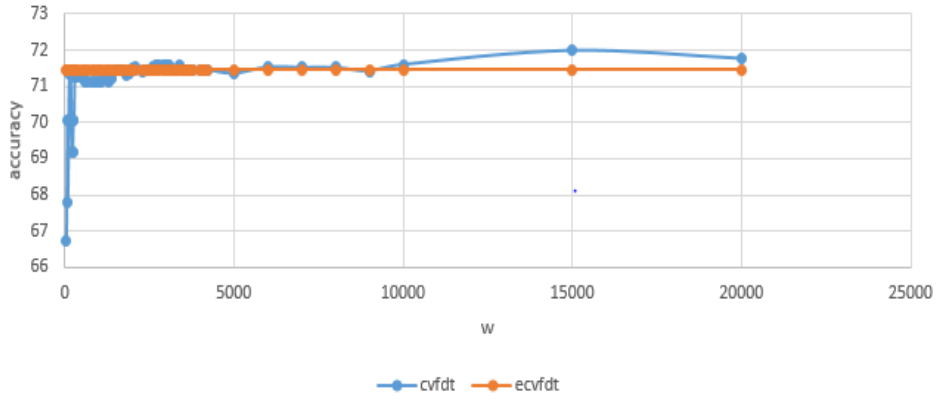


Fig8 Showing the comparision of Hyperpane1 for ACVFDT and E-CVFDT

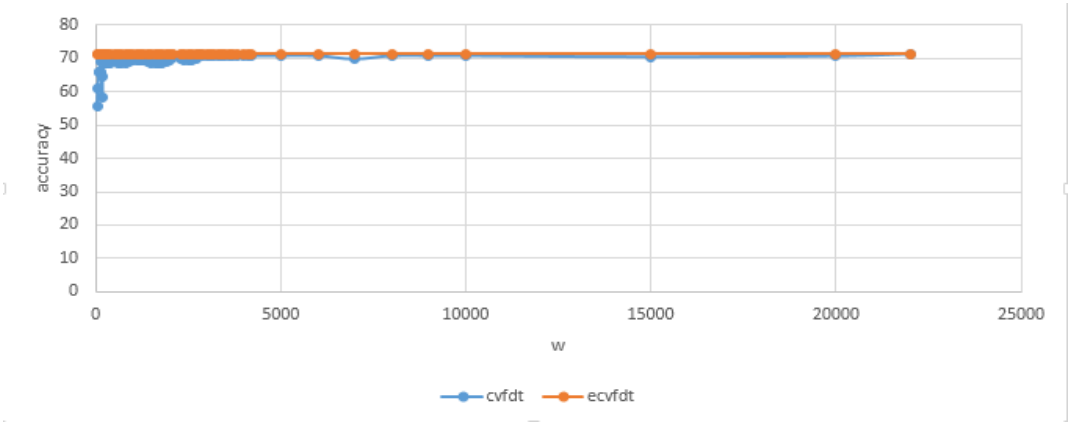


Fig9 Showing the comparision of Hyperpane4 for ACVFDT and E-CVFDT

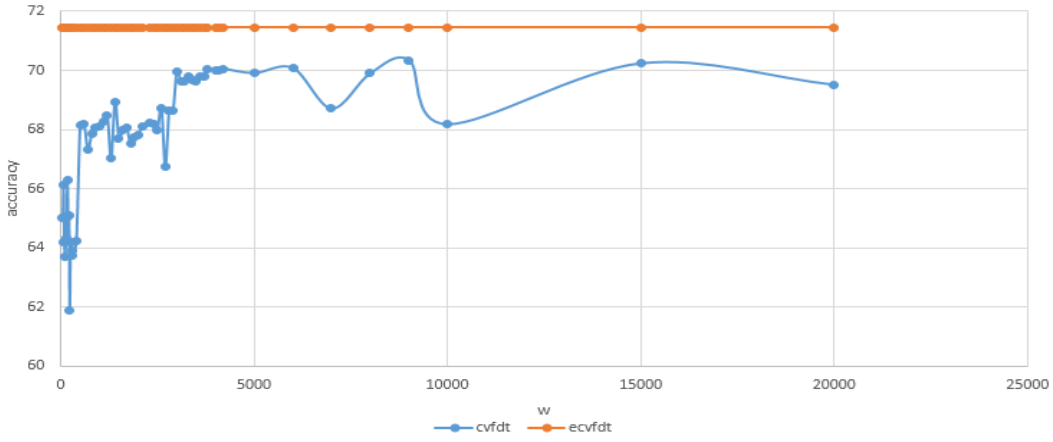


Fig10 Showing the comparision of Hyperpane1 for ACVFDT and E-CVFDT

5. Novel class detection using unsupervised learning

Mining Concept Drift from Data Streams by Unsupervised Learning is only the first step towards finding the Concept Drift for web based applications. As it is web-based it classifies the records over the web & helps to find the drift in constantly changing Streams.

5.1 Use of Unsupervised Learning

Many methods already exist for Concept Drift detection using Supervised Learning. Also issue with Supervised Learning is on detection of Concept Drift, it is difficult to predict if it gives rise to a Novel Class Label, New or Previously unknown Class Label. It is so because Supervised Learning goes with the assumption of pre-defined and known class labels. In the context of supervised learning each data is associated with a given class, which the algorithm must learn to predict, several solutions have been proposed for the classification of data streams in the presence of concept drift. These solutions are generally based on adaptive maintenance of a discriminatory structure, for example using a set of binary rules, decision trees or ensembles of classifiers. Also with supervised learning the issue of Window size comes up, as the maximum number of training set examples for each iteration can be equal to only the Window size. In unsupervised learning such issues don't exist, i.e. the Novel class because Unsupervised Learning doesn't start the learning process by a pre-defined set of known classes but forms the classes from the similarity/dissimilarity measures between training set examples. Also in Unsupervised Learning there is no concept of window, so the number of training examples to be taken in each iteration depends on the algorithm and not on anything else. It is better because in initial iterations there may be a need to take all the training set examples for clustering but in further steps it may be desired to reduce the training set examples, limited only to the ones which are not yet clustered properly.

The experimentation done was for the SEA Drift Set Database [13], which contains 50,000 records and 40% drift. SEA is an artificial dataset contains abrupt concept drift, first introduced in [13]. It is generated using three attributes, where only the two first attributes are relevant. All three attributes have values between 0 and 10. The points of the dataset are divided into 4 blocks with different concepts. In each block, the classification is done using $f1 + f2 \leq \theta$, where f1 and f2 represent the first two attributes and θ is a threshold value. The most frequent values are 9, 8, 7 and 9.5 for the data blocks.

Table6 Comparison of Results for different Learning Rates

Lear -ning Rate	Total Class 0	Total Class 1	Correc t Class 0 (TP)	TP %	Error Class 0	Diff Class 0 (ACT-CORR) (FP)	FP %	Correct Class 1 (TN)	TN %	Error Class 1	Diff Class 1 (ACT-CORR) (FN)	FN %
0.1	24568	25497	19598	63.92	4970	11061	36.08	14397	74.44	11100	4944	**25.56**
0.2	24961	25039	19510	63.64	5451	11149	36.36	13890	71.82	11149	5451	**28.18**
0.3	25354	24645	19439	63.4	5915	11220	36.6	13426	69.42	11220	5915	**30.58**
0.4	26538	23462	20148	65.72	6390	10511	34.28	12951	66.96	10511	6390	**33.04**
0.5	25875	24125	19262	62.83	6613	11397	37.17	12728	65.81	11397	6613	**34.19**
0.6	26390	23610	19513	63.65	6877	11146	36.35	12464	64.44	11146	6877	**35.56**
0.7	29354	20646	21433	69.91	7921	9226	30.09	11420	59.05	9226	7921	**40.95**
0.8	29809	20191	21789	71.07	8020	8870	28.93	11321	58.53	8870	8020	**41.47**
0.9	29883	20117	22000	71.76	7883	8659	28.24	11458	59.24	8659	7883	**40.76**
1.0	30134	19866	22536	73.51	7598	8123	26.49	11743	60.72	8123	7598	**39.28**

As it can be inferred from the results shown above that the Concept Drift detection which is the False Negative (FN) percentage shown in the Table 2, increase with the increase in the learning rate steadily from 0.1 to 1.0. Thus it can be inferred that with the increase in the learning rate we have more drift detection & the maximum drift which we find is at the learning rates of 0.7 & 0.9, as in the dataset [13] too it is mentioned that the drift present in the data is about 10% for each concept, which amounts to about 40% drift in the data. Thus it can be concluded that the above mentioned method is successful in the Drift Detection and new class labels which refers to novel class detection.

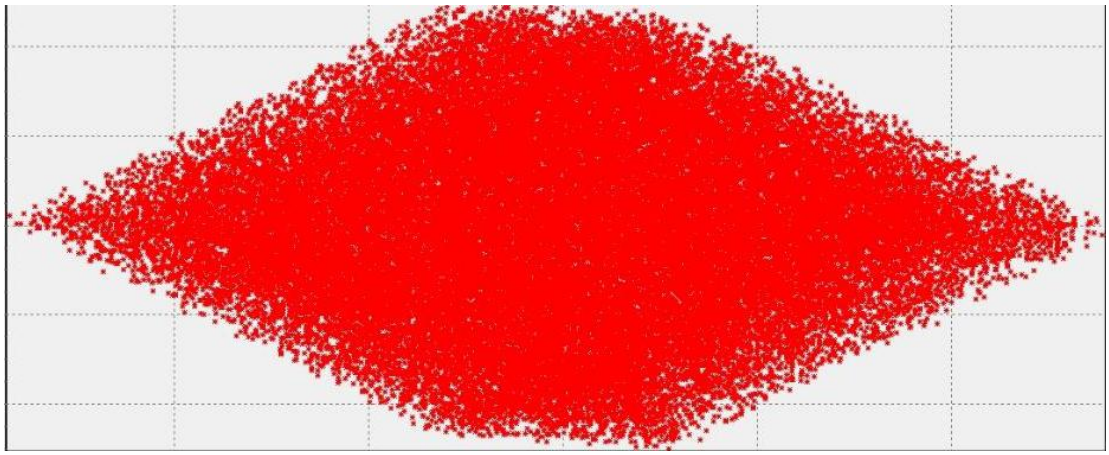


Fig11. Scatter Plot showing the SOM Model for the Dataset with 0.1 Learning Rate

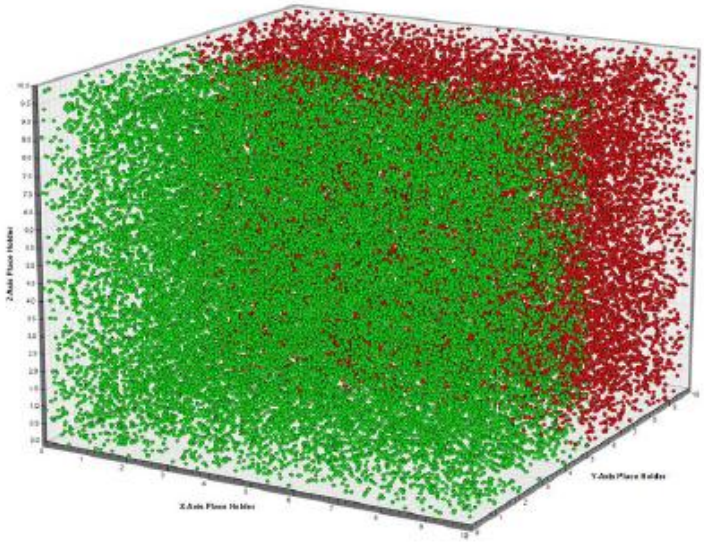


Fig12 Scatter Plot showing the records with New Class Labels for Learning Rate 0.1

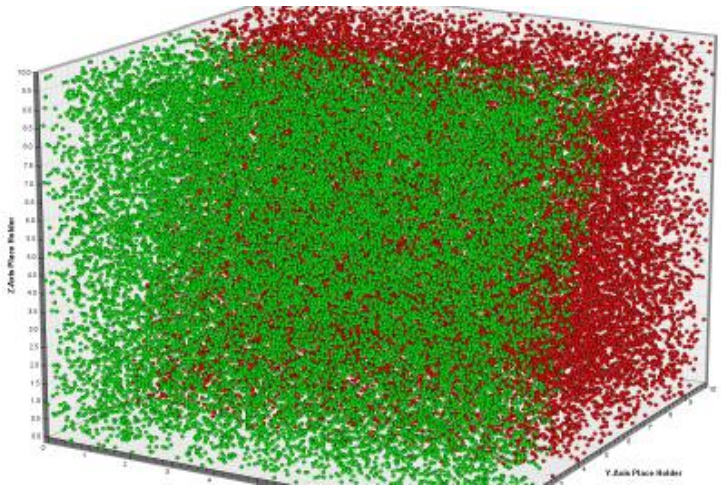


Fig13. Scatter Plot showing the records with New Class Labels for Learning Rate 10

6.CONCEPT-DRIFTING DATA STREAMS USING OPTIMIZED GENETIC ALGORITHM

The Data streams are extremely large and cannot be fully stored in the memory. They also have a peculiar time varying characteristic called concept drift [14]. Concept drift[15] is a phenomenon where the characteristics of the data stream changes due to the change in the underlying context. Due to the effect of concept drift, the accuracy of the model built for classifying the data stream will degrade and hence the model should be frequently upgraded and corrected in a fast manner in order to improve its predictive capability. So the algorithms designed to mine the data streams must have two important characteristics, namely adaptable learning and scalability.. The proposed GA based algorithm[16,17] is for mining data streams which is scalable and adaptable to concept drifts.

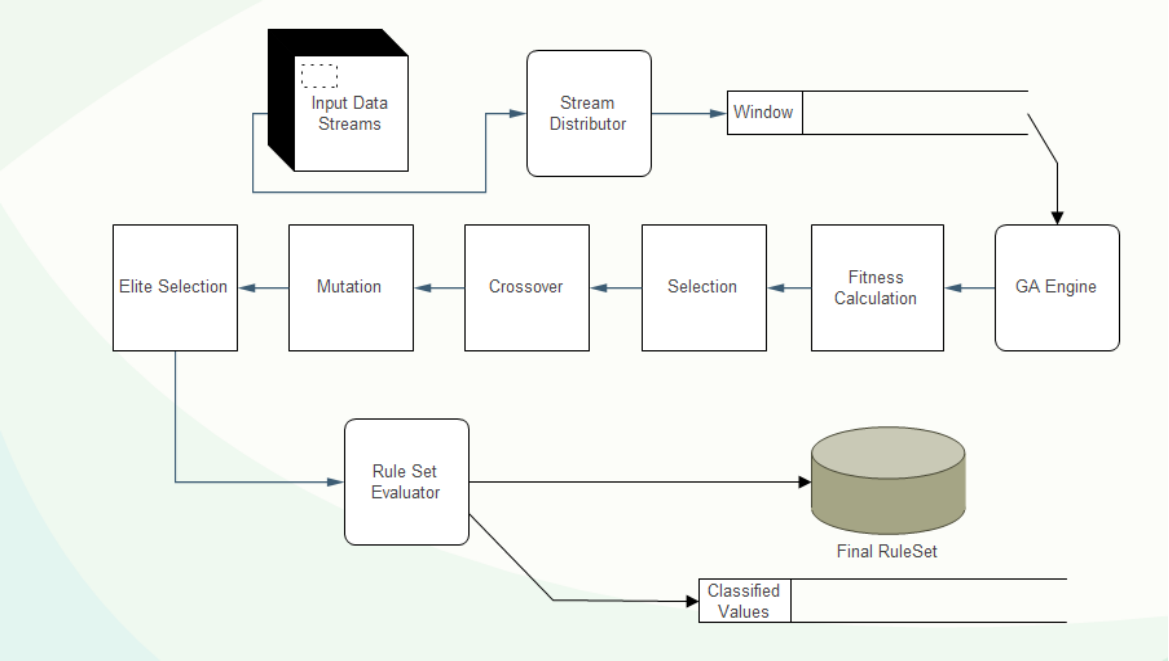


Fig14.Design Flow of OGA Process

There are four major functional units in the proposed method as described in the Fig14. They are
Data stream distributor
Population creator
Genetic engine
Rule set Evaluator

Algorithm Find Rule( )

1. For each Sub Population IR1 to IRn do
    a. Calculate fitness of each subpopulation
    b. Create a new sub population by applying following steps
    1. [Selection] Select two parent chromosomes from the population IRi based on their fitness
    2. [Crossover] With a crossover probability cross over the parents to form new offspring
    3. [Mutation]With a mutation probability mutate the new offspring .
    4. [Insert and Remove] Apply Insert and remove operator to the offspring.
    c. Replace the sub population IRi with the corresponding new child population.
2. Again repeat from step 1 [18].

Fitness calculation

The fitness of the rules in the population is calculated based on a function containing two terms namely Predictive Accuracy and Comprehensibility. A very simple way to measure the predictive Accuracy is

Predictive Accuracy $(PA) = (|A\&C| - 1/2)/|A|$ (1)

The standard way of measuring Comprehensibility is to count the number of conditions in the rule. If a rule has at most L conditions, the Comprehensibility of the rule (or individual) p can be defined as

Comprehensibility $(CM) = (L - x)/(L - 1)$. (2)

where x is number of attributes that take part in the corresponding rule. The fitness function is computed as the arithmetic weighted mean of Comprehensibility and Predictive Accuracy.

Fitness $= W1 * PA + W2 * CM$ (3)

W1 andW2 are weights assigned by user and their value depends on the user requirements (Generally W1=0.6 and W2=0.4). The GA Engine uses the current data in the Windows of the Data Distributor to calculate the fitness of its chromosomes.

Elite selection

Best Elite percent (normally range from 5 to 10%) of chromosomes of the population of the previous generation are considered as elite and they are copied to the next generation unaltered. After generating the new child population for all the classes, the GA Engine replaces all other chromosomes except the elite ones in the candidate rule sets IR1 to IRn of all the classes by the best chromosomes of their corresponding child population.

Rule set evaluator

It frequently scans the candidate rule sets IR1 to IRn and copies the best rule to the final rule set. Some rules in the final rule set may become inappropriate due to the concept drift. The rules whose support and confidence falls below a certain threshold continuously for certain number of generations are removed to make the classifier to adapt itself to the concept change. If two rules of the final rule set overlap and contradict each other, the latest rule is retained and the old rule is discarded. So if there is a concept drift the rules reflecting the concept drift will be generated naturally and will be added to the final rule set and at the same time rules pertaining to the previous concepts are removed gradually from the final rule set. Thus the final rule set contains rules reflecting the characteristics of the latest concepts of the data stream.

6.1 OGA Process with Datasets

Considering car data sets, which contain 1728 records and 6 attributes, all attributes are categorical. The target class attribute has four values namely ‘unacc’, ‘acc’, ‘good’, ‘vgood’. To generate larger data sets of size 10000, 20000 and 30000 the records are duplicated and randomly arranged such that the data distribution is proportionately similar to the original data set.

Attributes	Values
• Buying	vhigh, high, med, low.
• Maintenance	vhigh, high, med, low.
• Doors	2, 3, 4, 5more.
• Persons	2, 4, more.
• Lug boot	Small, med, big
• Safety	Low, med, high.

Now here in OGA Process,

Creation of Population is duplicating the records with size say suppose 1000 set of records from training data sets.

Individuals are the sets of records. Here in car data set, the example for individual is,

vhigh, vhigh, 3, 2, small, high, unacc

Chromosomes are the combination of target class and individual for generating the solution. Example for chromosome is,

target: acc and vhigh, med, 3, more, med, med

Genes are the solutions found after generating the solution in GA process with assigned target class label value. Example for genes isf`,

vhigh vhigh 3 2 small high unacc

Fitness value of an individual is the measure value of the fitness function for that individual. Here, fitness value is initiated with a minimum threshold value based on the best elitism selection.

Now, the **OGA Process for car datasets** is done in the following steps:

target class attributes unacc, acc, good and vgood is considered as 1000, 0100, 0010 and 0001.

Similarly for the other attributes,

Attributes	Values
• Buying	vhigh-1000, high-0100, med-0010 and low-0001.
• Maintenance	vhigh-1000, high-0100, med-0010 and low-0001.
• Doors	2-1000, 3-0100, 4-0010 and 5more-0001.
• Persons	2-1000, 4-0100 and more-0010.
• Lug boot	Small-1000, med-0100 and big-0010.
• Safety	Low-1000, med-0100 and high-0010

Now for example, the individuals
vhigh vhigh 3 2 small high unacc
Is considered as,

1000 1000 0100 1000 1000 0010 1000

Chromosomes are formed with the target class attribute for unacc-1000 and the individual for generating the solution. So total 7 attributes and 4 target class attributes forms 28 chromosomes.

Similarly, for all the rules genes solution set is generated.

The same OGA process is applied for other datasets also.

6.2 **Performance Evaluation Using Rival Algorithms**

Considering

I. Error Rate which is equal to the ratio of incorrectly classified values and 100

II. Classification Run Time.

Table 7 Classification Run Time (Seconds) tabulated using Different Classifications for 10 different datasets[19]

Index	Dataset	EC (Random Forest)	RBC (PART)	CVFDT	Optimized GA
1	KDDCup	57.33	164.58	131.84	0.01811
2	Car	598.5	0.03	1	0.00084
3	Chess	5.03	103.35	2	0.001315
4	Nursery	6.021	0.15	2	0.00087
5	Hyperplane	40.32	0.14	2	0.015452
6	Sea	68.78	0.12	1	0.004276
7	Letter	26.7	0.1	1.5	0.01417
8	Image Segmentation	0.18	0.01	0.07	0.000724
9	Solar Flare	0.13	0.01	0.04	0.000892
10	Yeast Database	1.48	0.02	0.49	0.003023

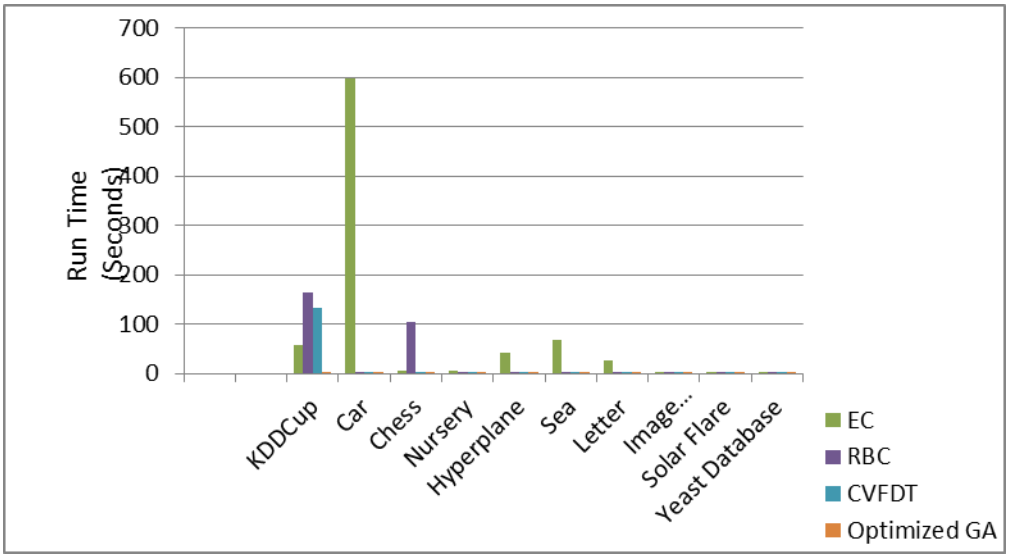


Fig15.Comparison of Classification Run Time (Seconds) using Different Classifications for 10 different datasets[19].

Table8 Error Rates tabulated using Different Classifications for 10 different datasets[19].

Index	Dataset	EC (Random Forest)	RBC (PART)	CVFDT	Optimized GA
1	KDD Cup	0.003	0.002375	0.15	0
2	Car	0.251	0.29978	0.29978	0
3	Chess	0.133412	0.122407	0.916844	0.001
4	Nursery	0.125	0.666667	0.498302	0.0015

5	Hyper plane	0.235	0.5378	0.4531	0
6	Sea	0.0895	0.4744	0.3741	0. 2
7	Letter	0.1389	0.98	0.603	0.49269
8	Image Segmentation	0.0435	0.95	0.0823	0.001
9	Solar Flare	0.168	0.1456	0.1183	0.001
10	Yeast Database	0	0.425202	0.38814	0

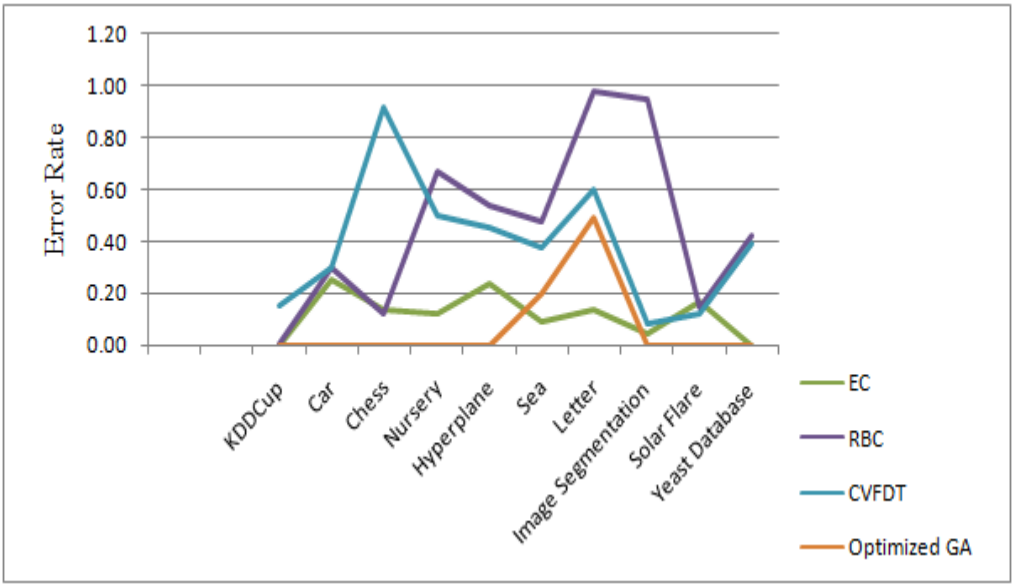


Fig16.Comparison of Error Rates using Different Classifications for 10 different datasets

Rivals Algorithm

To compare the algorithms' performance, error rate and run time of data sets are calculated. A win/lose/tie (w/l/t) record is calculated for each pair of the method for which the experiment is performed.

It represents the number of data sets in which an algorithm, respectively wins, loses or ties when compared with the other algorithm regarding error rate. Same is calculated for all algorithms with respect to run time. From that we can prove which algorithm has best performance.

Table 9 Performance Evaluation Using Rival Algorithm's w/l/t records with regard to their run time across 10 datasets[19].

Method	EC	RBC	CVFDT	Optimized GA
EC	0/0/10	2/8/0	1/9/0	0/10/0
RBC	**8/2/0**	0/0/10	**8/2/0**	0/10/0
CVFDT	**9/1/0**	2/8/0	0/0/10	0/10/0
OGA	**10/0/0**	**10/0/0**	**10/0/0**	0/0/10

Table 10 Performance Evaluation Using Rival Algorithm's w/l/t records with regard to their error rates across 10 datasets

Method	EC	RBC	CVFDT	Optimized GA
EC	0/0/10	**7/3/0**	**9/1/0**	2/7/1
RBC	3/7/0	0/0/10	2/7/1	0/10/0
CVFDT	1/9/0	**7/2/1**	0/0/10	0/10/0
OGA	**7/2/1**	**10/0/0**	**10/0/0**	0/0/10

Hence Optimized GA has highest winning probability from both classification error rate and run time which proves the best efficiency.

7. Concept Drift and Class Imbalance in Multi-Label Stream Classification

Many traditional learning systems are not prepared to induce a classifier that accurately classifies the minority class under such situation. Frequently, the classifier has good classification accuracy for the majority class, but its accuracy for the minority class is unacceptable. The problem arises when the misclassification cost for the minority class is much higher than the misclassification cost for the majority class. Unfortunately, that is the norm for most applications with imbalanced data sets, since these applications aim to profile a small set of valuable entities that are spread in a large group of "uninteresting" entities.

Classification methods for unbalanced are currently considered at two levels. At the data level, imbalance can be eliminated or reduced by changing the data distribution. Most algorithms can be used to resolve data using 2 approaches: over-sampling and under-sampling. The first method increases minority class samples to improve classification performance of the minority class. The easiest method is to simply copy the minority class sample. This method leads to the natural introduction of additional training data and increases the training time, but does not add useful information to the sample, eventually leading to over-fitting. However, the synthetic minority over-sampling technique (SMOTE)[20] , a machine-learning approach based on over-sampling theory, could be used to avoid over-fitting but may introduce noise. Another method of reducing imbalances is by reducing the size of the majority class. This can be accomplished by randomly removing some of the samples in the majority class, which can lead to the loss of useful information.

7.1 Data level balancing techniques

Ensemble and Balance cascade

Xu-Ying Liuet [21] has proposed two ensemble techniques known as Easy Ensemble and Balance Cascade. In the first proposed approach several subsets samples from the majority class are prepared and trains a learner using each of them, and combines the outputs of those learners. The second proposed approach trains the learners sequentially, where in each step, the majority class exam7ples that are correctly classified by the current trained learners are removed from further consideration.

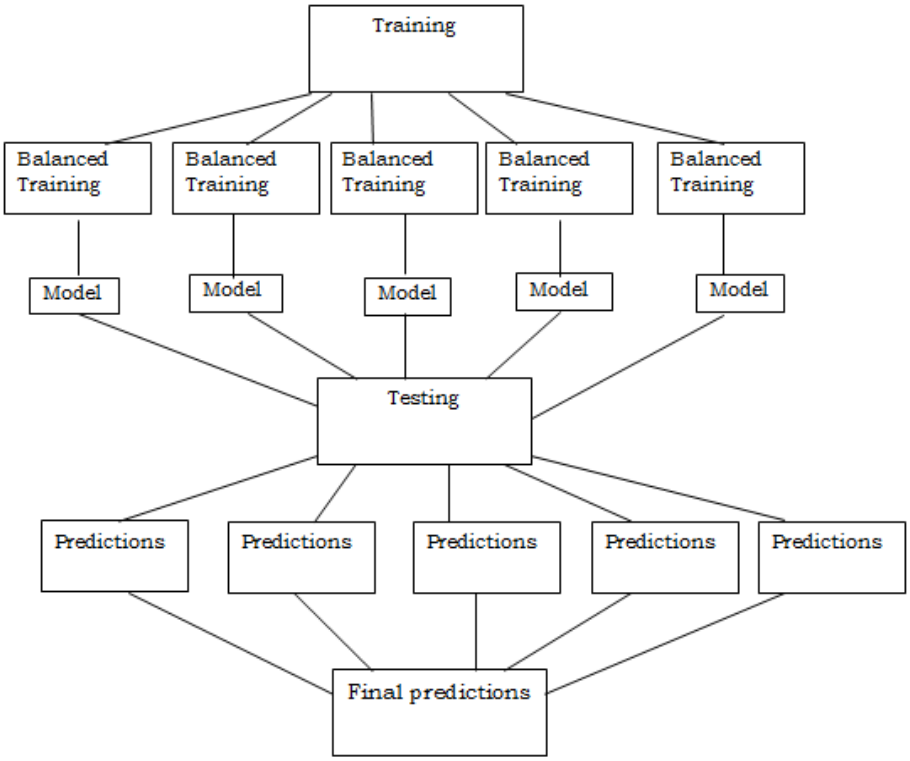


Fig17.Easy Ensemble [16]

Easy Ensemble learns different aspects of the original majority class in an unsupervised manner.

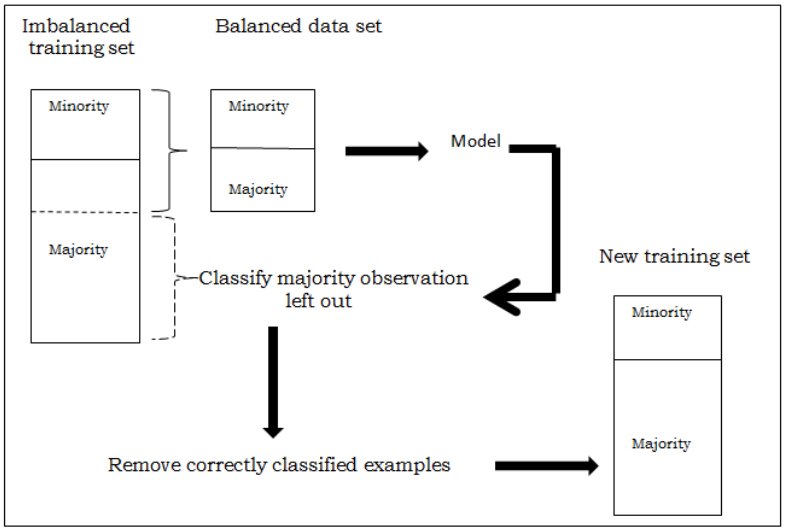


Fig18.Balance cascade method [22]

Balanced cascade method keep removing majority class examples until none is miss classified, it explores the majority class in a supervised manner.

7.3 Algorithm level –cost sensitive learning

A classifier induced from an imbalanced data set has, typically, a low error rate for the majority class and an unacceptable error rate for the minority class. The problem arises when the misclassification cost for the minority class is much higher than the misclassification cost for the majority class. In this situation, it is important to accurately classify the minority class in order to reduce the overall cost. A cost-sensitive learning system can be used in applications where the misclassification costs are known. Cost-sensitive learning systems attempt to reduce the cost of misclassified examples, instead of classification error.

Depending on the AUC measurement, the performance of each sampling technique was investigated. Glass data set is selected from Keel imbalanced data sets this data set is addressing imbalance and multi class concepts ,it is tested against some of the oversampling methods and under sampling methods with the cost-sensitive learning(C_SVMCS) and Ensemble model(Easy ensemble). After comparing the two methods best combination of both data preprocessing models and classifiers for the given data set are SMOTE_TL, SMOTE_ENN. The combination of SMOTE_NN with Easy Ensemble improves the accuracy and the SMOTE_TL with cost –sensitive learning [23] will out perform well in the imbalanced data classification.

7.4Concept drift in class imbalance multi label stream

The proposed approach Multiple Windows deals with the existing challenges of MLSC [24]. This new Multiple Windows (MW) method maintains two fixed-size windows per label, one for positive and one for negative examples. This is accomplished in a space-efficient way through instance-sharing between windows by a time-efficient instantiation using k-Nearest-Neighbors (kNN) as the base classifier for each label. Class imbalance is further tackled using a new batch-incremental thresholding technique that accurately translates the probabilistic estimates for each label to bipartitions.

The proposed method for handling concept drift with support vector machines method directly implements the goal of discarding irrelevant data with the aim of minimizing generalization error. Exploiting the special properties of SVMs, it estimates the window size selection problem. Unlike for the conventional heuristic approaches, this gives the new method a clear and simple theoretical motivation. Furthermore, the new method is easier to use in practical applications, since it involves less parameters than complicated heuristics. Experiments in an information altering domain show that the new algorithm achieves a low error rate and selects appropriate window sizes over very different concept drift scenarios. An open question is how sensitive the algorithm is to the size of individual batches. Since in the current version of the algorithm the batch size determines the estimation window, the variance of the window size is likely to increase with smaller batches. It might to be necessary to select the estimation window size independent of the batch size. A shortcoming of most existing algorithms handling concept drift (an exception is Lanquillon (1999)) is that they can detect concept drift only after labeled data is available. That is, after the learning algorithm starts making mistakes. While this app ears unavoidable for concept drift with respect to Pr (y j ~ x), it might be possible to detect concept drift in Pr(~ x) earlier by using transductive support vector machines.

Multi-label classification is a challenging and appealing supervised learning problem where a sub- set of labels, rather than a single label seen in traditional classification problems, is assigned to a single test instance. Classifier chains based methods are a promising strategy to tackle multi-label classification problems as they model label correlations at acceptable complexity. However, these methods are difficult to approximate the underlying dependency in the label space, and suffer from the problems of poorly ordered chain and error propagation. In this paper, a novel poly tree-augmented classifier chains method to remedy these problems. A poly tree is used to model reasonable conditional dependence between labels over attributes, under which the directional relationship between labels within causal basins could be appropriately determined.

Unlike traditional single label classification problems where an instance is associated with a single-label, multi-label classification (MLC) attempts to allocate multiple labels to any input unseen instance by a multi-label classifier learned from a training set. Obviously, such a generalization greatly raises the difficulty of obtaining desirable prediction accuracy at a tractable complexity. Nowadays, MLC has drawn a lot of attentions in a wide range of real world applications, such as text categorization, semantic image classification, music emotions detection and bioinformatics analysis. A convenient and straightforward way for MLC is to conduct problem transformation in which a MLC problem is transformed into one or more single label classification problems.

For experimentation two data sets . reuters 21578 and newsgroup were used and by considering the Area under curve the performance of the proposed method is plotted in Fig19,20. Reuters news documents It contains 21578. They were labelled manually by Reuters personnel. Labels belong to 5 different category classes, such as 'people', 'places' and 'topics'. The total number of categories is 672, but many of them occur only very rarely. He presented the format in 22 files of 1000 documents delimited by SGML tags. The size of the reuters 21578 dataset is 27MB.Coming to newsgroup dataset, this is a well-known data set for text classification, used mainly for training classifiers by using both labelled and unlabelled data. The data set is a collection of 20,000 messages, collected from UseNet postings over a period of several months in 1993.Many of the categories fall into overlapping topics; for example 5 of them are about companies' discussion groups and 3 of them discuss religion. Other topics included in News Groups are: politics, sports, sciences and miscellaneous.

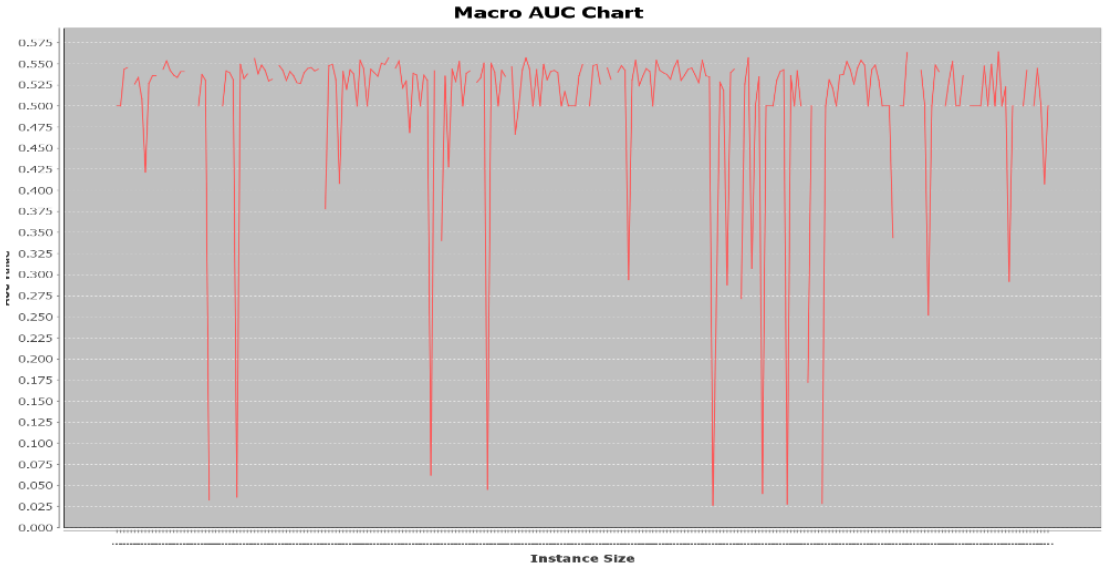


Fig19.Performance curve for News group data set for MLC method

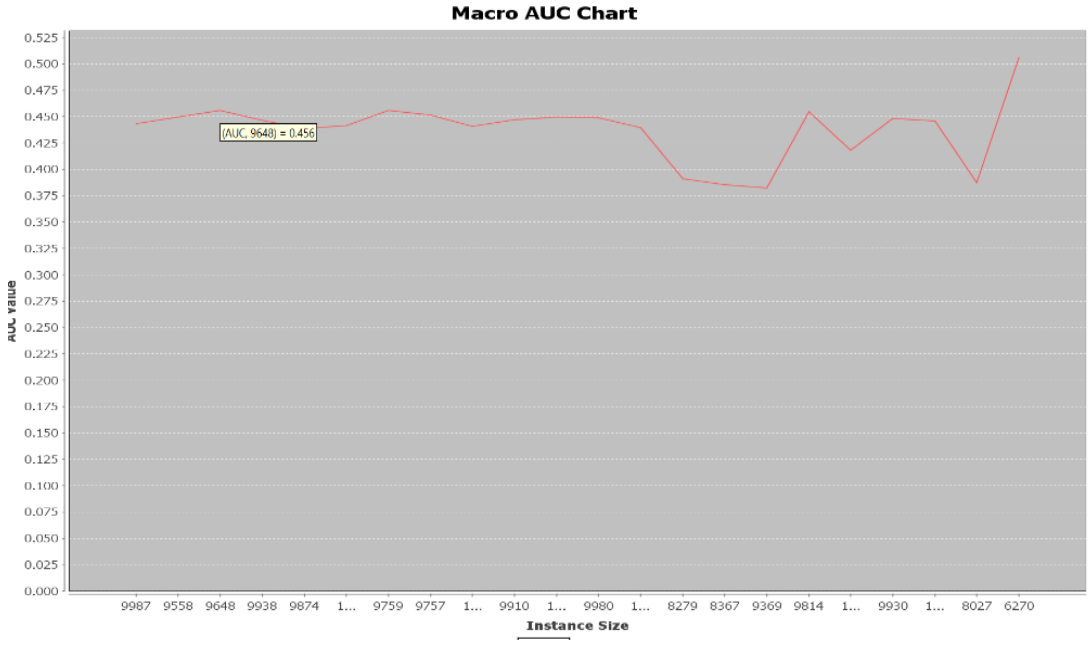


Fig 20. Performance curve for Reuters 21578 data set for MLC method

8. Discussion of the result

Ensemble Classifiers are used to get high accuracy in the classification of the data streams. In the proposed method the combination of the bagging and the Random forest are giving the high accuracy. Bagging and Random Forest are two methods which transform the "weak "individual models in a "strong" ensemble of models. Ensemble models can be used in novel class detection in concept drifting data streams with this misclassification error can be minimized in the concept drifting classification.

The decision tree algorithms Hoeffding tree,VFDT, CVFDT, E-CVFDT are used for the classification of data. The Hoeffding tree has a drawback of using large memory and stacking up the old instances. VFDT algorithm overcomes some of the drawbacks of Hoeffding tree because it can differentiate ideal attribute and wasteful attribute but it still cannot handle concept drift. CVFDT algorithm handles concept drift by using window mechanism and also increases efficiency by generating alternate sub-tree but the efficiency is reduced because of its general method of handling instances in CVFDT without considering the types of concept drift.The proposed method AgentbasedCVFDT increased the accuracy when compared to VFDT.Further drawback of CVFDT is addressed by E-CVFDT. With the help of test bed dataset rotating hyperplane accuracies of ACVFDT and E-CVFDT are compared.

Comparison of single classifier and ensemble agent proved that proposed technique increases the accuracy. Multi agent system classifier Efficiency depends on the number of partitions, too many partitions and too less partitions will decrease the accuracy of the decision system. There is no hard and tough rule to decide the number of partitions .This proposed system is not dependent on application and not limited to specific datasets, it can be implemented in various domains where mining of knowledge plays major role and accuracy is crucial.

The data streams are not stored fully in any of the earlier classification techniques due to their concept drift. Optimized GA is such a technique where the classification is done for concept-drifting data streams by using streaming window and its mechanisms like selection, crossover, mutation and elitism for the generation of the solution with best fitness value for best classification rate. Further, the OGA can be optimized by minimizing the build time for construction of the model for even large data sets when streamed enhances performance and time efficiency.

A dataset for modeling is perfectly balanced when the percentage of occurrence of each class is 100/n, where n is the number of classes. If one or more classes differ significantly from the others, this dataset is called skewed or unbalanced.

The unbalanced data always affects the accuracy of the system. This is addressed in this work at data level and algorithm level.

Table11. Comparison of the proposed learning methods

Learning Method	Concept drift	Memory required	Accuracy
Ensemble Agent	Efficient	More	89%
Agent Supervised Method	Efficient	More	79.8
Novel class detection Method	Efficient	Less	75%
Optimized Genetic method	Efficient	Less	82%

9. Conclusion

In this work all the learning methods are discussed with balanced data sets and unbalanced data sets .In Ensemble technique cross validation technique is used. As the cross validation method gives the accurate results, each partition of the data set will be classified by all agents with different classifiers which yield good accuracy. The obtained results are compared with the other single classifiers and the high accuracies are proved.

In supervised method ACVFDT and E-CVFDT are implemented. In ACVFDT drift is found by Agents and classifies, with E-CVFDT it is proved that all types' drifts can be identified. In Unsupervised method along with the Concept detection Novel class detection also implemented. Optimized genetic algorithm is implemented and compared with other methods and it is proved it is good for large data sets.

Imbalanced data sets are considered for classification which is taken from KEEL software .Classification is performed both at data level and algorithm level. MLSC is implemented with two windows to favor the positive class.

References

[1]G. Hulten, L. Spencer, and P. Domingos , Mining Time-Changing Data Streams,‖ Proc. *Seventh ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining* (KDD '01), pp. 97-106, 2001.

[2] D. J. Newman, S. Hettich, C. L. Blake, and C. J. Merz. UCI repository of machine learning databases, 1998.

[3] C.X. Ling and V.S. Sheng. Cost-sensitive learning and the class imbalance problem. Encyclopedia of Machine Learning, 2008.

[4]Pedro Domingos, Geoff Hulten, " Mining High Speed Data Streams",KDD-00 in proceeding of sixth ACM SIGKDD international conference on knowledge discovery and data mining, USA, 2000, pp 71-80.[5]Leo Breiman (2001). Random orests.Machine Learning. 45(1):5-32.

[6]J.C.Schimmer and R.H.Ganger Beyond incremental processing :Tracking Concept Drift .In proceedings of the fifth National conference on Artificial Intelligence .pages 502-507 AAAI press ,Menlo park ,CA,1986.

[7] N. Japkowicz and S. Stephen. The class imbalance problem: A systematic study. Intelligent data analysis,6(5):429{449, 2002.

[8] W. Nick Street and Yong Seog Kim. A Streaming Ensemble Algorithm (SEA) for Large- Scale Classification. *KDD – 01*. San Francisco, CA.

[9] D. J. Newman, S. Hettich, C. L. Blake, and C. J. Merz. UCI repository of machine learning databases, 1998.

[10]. Dougherty, J., R. Kohavi, and M. Sahami. Supervised and unsupervised discretization of continuous features. In Proceedings of International Conference on Machine Learning (ICML-1995), 1995.

[11] Pedro Domingos, Geoff Hulten, " Mining High Speed Data Streams",KDD-00 in proceeding of sixth ACM SIGKDD international conference on knowledge discovery and data mining, USA, 2000, pp 71-80.

[12] A. Tsymbal. "The problem of concept drift: definitions and related work", Technical Report TCD-CS-2004-15, Computer Science Department, Trinity College Dublin, Ireland. 2004.

[13] W. Nick Street and Yong Seog Kim. A Streaming Ensemble Algorithm (SEA) for Large- Scale Classification. *KDD – 01*. San Francisco, CA.

[14] E Padmalatha, C R K Reddy and Padmaja B Rani. Article: Ensemble Classification for Drifting Concept. *International Journal of Computer Applications* 80(11):33-36, October 2013.

[15] E.Padmalatha,C.R.K.Reddy, B.Padmaja Rani "Classification of Concept Drift Data Streams"In the proceedings of the Fifth International Conference on Information Science and Applications .ICISA 2014.IEEE PP291-295, 2014.

[16] Periasamy Vivekanandan and Raju Nedunchezhian, "Mining data streams with concept drifts using genetic algorithm", Artificial Intelligence Review, Vol. 36, Issue 3, pp 163-178, Springer, October 2011.

[17] Basheer M. Al-Maqaleh and Hamid Shahbazkia, "A Genetic Algorithm for Discovering Classification Rules in Data Mining", International Journal of Computer Applications (0975-8887), Vol. 41-No. 18, March 2012.

[18] Syed Shaheena and Shaik Habeeb, "Classification Rule Discovery Using Genetic Algorithm-Based Approach", NIMRA Institute, Department of CSE, IJCTT Journal, Vol. 4, Issue 8, pp 2710-2715, August 2013.

[19] E Padmalatha, C R K Reddy and Padmaja B Rani. Article: Classification of Concept-Drifting Data Streams using Optimized Genetic Algorithm. *International Journal of Computer Applications* 125(15):1-6, September 2015.

[20]Wei Liu, Sanjay Chawla, David A. Cieslak, Nitesh V. Chawla, ― A Robust Decision Tree Algorithm for Imbalanced Data Sets‖, 2010.

[21]Xu-Ying Liu, Jianxin Wu, Zhi-Hua Zhou" Exploratory Undersampling for Class-Imbalance Learning", IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, VOL. 39, NO. 2, APRIL 2009, pp.no:539 – 550.

[22] X.Y. Liu, J. Wu, and Z.H. Zhou. Exploratory undersampling for class-imbalance learning. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 39(2):539{550, 2009. [12]12Data Mining: Concepts and Techniques. J. Han and M. Kamber. Morgan Kaufmann, 2000.

[23] Junfeng Pan and Qiang Yang, Yiming Yang and Lei Li, Frances Tianyi Li and George Wenmin Li "Cost-SensitiveData Preprocessing for Mining Customer Relationship Management Databases", JANUARY/FEBRUARY 2007, A Technical Report.

[24] J. Read, B. Pfahringer, G. Holmes, and E. Frank. Classifier chains for multi-label classification. In *Proceedings of ECML PKDD '09*, pages 254–269, 2009.

[25]]Macskassy, S.A. and Provost, F.J., "Confidence Bands for ROC Curves," CeDER Working Paper 02-04, Stern School of Business, New York University, NY, NY 10012. Jan 2004.

