Mining Concept Drift from Data Streams by Unsupervised Learning

E.Padmalatha
Research scholar

C.R.K.Reddy, Ph
Professor

Padmaja Rani,Ph.D
Professor

ABSTRACT
Mining is involved with knowing the unknown characteristics from the databases or gaining of Knowledge (Knowledge Discovery) from Databases to get more useful information from the database. Real time databases which are constantly changing with time, there may arise a point when traditional Data Mining techniques may not be adequate as there may be a previously unknown class label involved or new properties of data which need to be taken into consideration. Thus as time passes and new data is in the dataset, the model predicted by the data mining techniques may become less accurate. This phenomenon is known as Concept Drift. The meaning of Concept Drift is the statistical properties of the target variable, i.e. how the properties of the target variable change over the course of time. The basic idea behind the ―Mining Concept Drift from Data Stream by Unsupervised Learning‖ is to detect the Concept Drift present in the Data Stream, which is used in majority of Web-Based Applications like Fraud Detection & Span E-mail Filtering etc. The approach taken here is both for the Offline Approach & an Online Approach, which can be easily merged with the current Web-Based Applications. Some examples for Concept Drift are – In a fraud detection application the target concept may be a binary attribute FRAUDULENT with values "yes" or "no" that indicates whether a given transaction is fraudulent. Or, in a weather prediction application, there may be several target concepts such as EMPERATURE, PRESSURE, and HUMIDITY. Each of these target parameters change over time and over model should be able to accommodate these changes or the Concept. In order to overcome the problems of the Offline or Desktop based processing to detect the Concept drift (which is available), it is aimed here to move the Concept Drift Detection process to the Cloud (web) & have it for Web-Based Applications too.

General Terms
SEA (Streaming Ensembling Algorithm), SOM

Keywords
Concept Drift, Data mining ,Data Stream.

1. INTRODUCTION
Traditional classification methods work on static data, and they usually require multiple scans of the training data in order to build a model [1]. The advent of new application areas such as ubiquitous computing, e-commerce, and sensor networks leads to intensive researchon data streams. In particular, mining data streams for actionable insights has become an important and challenging task for a wide range of applications [2].

For many applications, there are two major challenges in mining data streams:

- The data distributions are constantly changing and
- Most alerts monitored are rare occurring.

Clearly, the major challenge lies not in the tremendous data volume but, rather, in the concept drifts [3].

In classifying stream data with non-stationary class distribution, only the training phase is used to adjust the models. Without the feedback, there is no way to predict whether there is a concept shift in the underlying data. In reality, there is an investigation of a subset of testing cases to get their real label (for example, in a bank, certain transactionsare manuallyinvestigated). Because such investigations may take time, the labeled data may come with a lag. However, usually, this lag can be ignored.

Data streams pose several unique problems that make obsolete the applications of standard data analysis. Indeed, these databases are constantly online, growing with the arrival of new data. Thus, efficient algorithms must be able to work with a constant memory footprint, despite the evolution of the stream, as the entire database cannot be retained in memory. This may implies forgetting some information over time.

Another difficulty is known as the ―concept drift‖ problem: the probability distribution associated with the data may change over time. Any learning algorithm adapted to streams should be able to detect and manage these situations. In the context of supervised learning (each data is associated with a given class that 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 [4] or ensembles of classifiers [5], [6].

2. PROBLEM SPECIFICATION
In the data streams with the adding of data over time the model proposed for the data becomes less accurate giving rise to the problem of Concept Drift [7]. If the model which is being built doesn't take into consideration the Drift Factor while prediction, over time the outcomes of the model will become less reliable, and then the model will have to build from scratch & this process will continue. If the Drift factor is taken into consideration while the model is being built, then the built model will be more flexible and will help in predicting/classifying the stream in a better manner over time, even with the continuous addition of data. So it is proposed to find the Concept Drift in the Data Stream, by using Unsupervised Learning. The approach here deals with an unsupervised framework (class labels are unknown),

which requires adaptations to the presence of concept drift for the analysis of data streams. A method is proposed for synthetic representations of the data structure and a heuristic measure of dissimilarity between these models to detect temporal variations in the structure of the stream (concept drifts). The advantage of this method is the comparison of structures by means of models that describe them, allowing comparisons at any time scale without overloading the memory. Thus, it is possible to compare the structure of the stream in two potentially very distant time periods, since the models describing these periods can be stored in memory at very low cost.

2.1Unsupervised Learning

In machine learning,the problem of unsupervised learning is that of trying to find hidden structure in unlabeled data. Since the examples given to the learner are unlabeled, there is no error or reward signal to evaluate a potential solution. This distinguishes unsupervised learning from supervised learning and reinforcement learning.Unsupervised learning is closely related to the problem of density estimation in statistics [8]. However unsupervised learning also encompasses many other techniques that seek to summarize and explain key features of the data.
Many methods employed in unsupervised learning are based on data mining methods used to preprocess data**.** Approaches to unsupervised learning include:

- Clustering (e.g., k-means, mixture models, hierarchical clustering),
- Hidden Markov models,

Among neural network models, the self-organizing map (SOM) and adaptive resonance theory (ART) are commonly used unsupervised learning algorithms. The SOM is a topographic organization in which nearby locations in the map represent inputs with similar properties. The ART model allows the number of clusters to vary with problem size and lets the user control the degree of similarity between members of the same clusters by means of a user-defined constant called the vigilance parameter.

3. SELF ORGANIZING MAPS (SOM)

A self-organizing map (SOM) or self-organizing feature map (SOFM) is a type of artificial neural network (ANN) that is trained using unsupervised learning to produce a low dimensional (typically two-dimensional), discretized representation of the input space of the training samples, called a map.

Self-organizing maps are different from other artificial neural networks in the sense that they use a neighborhood function to preserve the topological properties of the input space.This makes SOMs useful for visualizing low-dimensional views of high-dimensional data, akin to multidimensional scaling. The model was first described as an artificial neural network by the Finnish professor TeuvoKohonen, and is sometimes called a Kohonen map or network [9]. Like most artificial neural networks, SOMs operate in two modes:

- Training - It builds the map using input examples (a competitive process, also called vector quantization)
- Mapping –It automatically classifies a new input vector

A self-organizing map consists of components called nodes or neurons. Associated with each node is a weight vector of the same dimension as the input data vectors and a position in the map space. The usual arrangement of nodes is a two-dimensional regular spacing in a hexagonal or rectangular grid. The self-organizing map describes a mapping from a higher dimensional input space to a lower dimensional map space. The procedure for placing a vector from data space onto the map is to find the node with the closest (smallest distance metric) weight vector to the data space vector. While it is typical to consider this type of network structure as related to feed-forward networks where the nodes are visualized as being attached, this type of architecture is fundamentally different in arrangement and motivation .It has been shown that while self-organizing maps with a small number of nodes behave in a way that is similar to K-means, larger self-organizing maps rearrange data in a way that is fundamentally topological in character.

It is also common to use the U-Matrix. The U-Matrix value of a particular node is the average distance between the node and its closest neighbors. In a square grid, for instance, the closest four or eight nodes (the Von Neumann and Moore neighborhoods, respectively) may be considered, or six nodes in a hexagonal grid. Large SOMs display emergent properties. In maps consisting of thousands of nodes, it is possible to perform cluster operations on the map itself as visible in Figure .1.

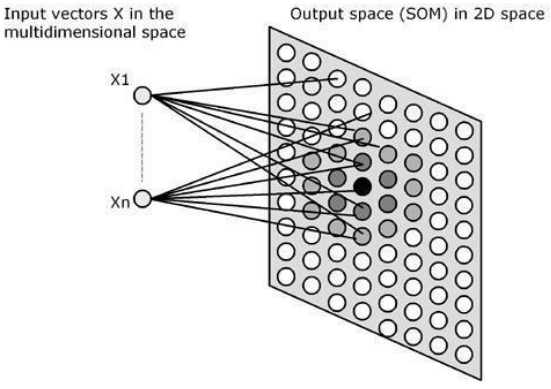

Figure1:Mapping of Inputs into a Self –Organizing Map

3.1 SOM Learning Algorithm Overview

A SOM does not need a target output to be specified unlike many other types of network. Instead, where the node weights match the input vector, that area of the lattice is selectively optimized to more closely resemble the data for the class the input vector is a member of. From an initial distribution of random weights, and over many iterations, the SOM eventually settles into a map of stable zones. Each zone is effectively a feature classifier, t h e graphical output is a type of feature map of the input space. Each

of the trained network by SOM, represent the individual zones. Any new, previously unseen input vectors presented to the network will stimulate nodes in the zone with similar weight vectors.

3.2 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 & 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.

4. IMPLEMENTATION DETAILS

It is proposed to implement the following by the use of Unsupervised SOM method to find the Concept Drift in the Data Streams. The methods followed for the same would be building of SOM from the datasets as specified, finding of the density function between the built SOM density models. From the models, finding the dissimilarity function to detect if drift is present or not in the Data Streams.

4.1 SOM Models

The SOM models would be built by the use of software by name of **Tanagra**. Tanagra is a free suite of machine learning software for research and academic purposes developed by Ricco Rakotomalala at the Lumière University Lyon 2, France. It is Open Source & supports all major data mining methods. It has options for SOM as required for specifying parameters etc. Also the algorithm described above is implemented in PHP (Personal Home Pages), for the concept to be implemented over the web..

4.2 Data Sets

SEA Concepts Dataset - Dataset (proposed by Street and Kim, 2001) [10] with 50,000 examples, three attributes and two classes. Attributes are numeric between 0 and 10, and all three are relevant. There are four concepts, 15,000 examples each, with different thresholds for the concept function, which is if relevant_feature1 + relevant_feature2 > Threshold then class = 0. Threshold values are 8, 9, 7, and 9.5. Dataset has about 10 % of noise.

So for the SEA dataset, the analysis is done on all the records. For each of the record a density & local neighborhood value is found & from them the new class label for the dataset according to the SOM Model. This is again stored in the database for different learning rates from 0.1 to 1.0. Then for each of the learning rates, a dissimilarity comparison is done with respect to the original dataset to find the drift present in the dataset according to the learning rates. For all the learning rates a comparative study is done & results are given.

Table 1. Description of full_db Table

Column	Datatype	Length	Precision	Scale	Primary key	Nullable	Default	comment
Sno	Integer	11	-	-	1	No	None	-
F1	Double	-	-	-	-	No	None	-
F2	Double	-	-	-	-	No	None	-
F3	Double	-	-	-	-	No	None	-
Class	Integer	11	-	-	-	No	None	-

Table2:Description of full_db_som_01 Table

Column	**Datatype**	**Length**	**Precision**	**Scale**	**Primary key**	**Nullable**	**Default**	**comment**
Sno	**Integer**	**11**	**-**	**-**	**1**	**No**	**None**	**-**
F1	**Double**	**-**	**-**	**-**	**-**	**No**	**None**	**-**
F2	**Double**	**-**	**-**	**-**	**-**	**No**	**None**	**-**
F3	**Double**	**-**	**-**	**-**	**-**	**No**	**None**	**-**
Class	**Integer**	**11**	**-**	**-**	**-**	**No**	**None**	**-**
Som_class	**Varchar**	**64**	**-**	**-**	**-**	**No**	**None**	**-**

4.3 Database Tables Descriptions Original Dataset Table

Table 4.1: Description of full_db Table

5. EXPERIMENTATION AND ESULTS

The experiments were carried out on the SEA Dataset [11], which has about 50,000 records with 3 attributes with the attribute values between 0 & 10. Each of the record of the Dataset is associated with a Class Label of 0 or 1. The experiment was carried out for the Learning rates of 0.1 to 1.0 & the results for each of them are as follows –

5.1 Learning Rate of 0.1 Actual Dataset Record Details Class 1 – 19341 Class 0 – 30659

Table3:Analysis for Learning rate 0.1

Learning Rate	0.1
TotalClass 0	24544
Total Class1	25481
Correct Class 0(TP)	19574
TP%	63.84
Error Class0	4970
Diff Class 0(ACT-CORR)(FP)	11085
FP%	36.16
Correct Class 1(TN)	14381
TN%	74.35
Error Class 1	11100
Diff Class 1(ACT-CORR)(FN)	4960
FN%	25.65
Learning Rate	0.2
Total Class 0	24961
Total Class1	25039
Correct Class 0(TP)	19510
TP%	63.64
Error Class0	5451
Diff Class 0(ACT-CORR)(FP)	11149
FP%	36.36
Correct Class 1(TN)	13890
TN%	71.82
Error Class 1	11149
Diff Class 1(ACT-CORR)(FN)	5451
FN%	28.18

Table4: Analysis for Learning rate of 0.2

Learning Rate	0.3
TotalClass 0	25354
Total Class1	24646
Correct Class 0(TP)	19439
TP%	63.4
Error Class0	5915
Diff Class 0(ACT-CORR)(FP)	11220
FP%	36.6
Correct Class 1(TN)	13426
TN%	69.42
Error Class 1	11220
Diff Class 1(ACT-CORR)(FN)	5915
FN%	33.04

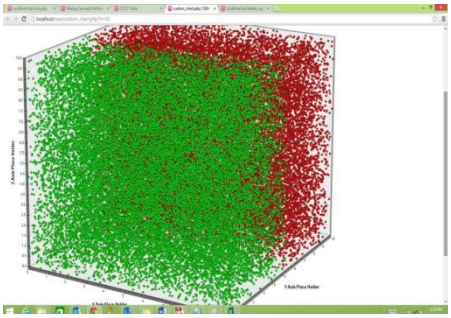

Figure2 :New Class lable for Learning Rate 0.1

Table5: Analysis for Learning rate of 0.3

Learning Rate	0.4
TotalClass 0	26538
Total Class1	23462
Correct Class 0(TP)	20148
TP%	65.72
Error Class0	6390
Diff Class 0(ACT-CORR)(FP)	10511
FP%	34.28
Correct Class 1(TN)	12951
TN%	66.96
Error Class 1	10511
Diff Class 1(ACT-CORR)(FN)	6390
FN%	33.04

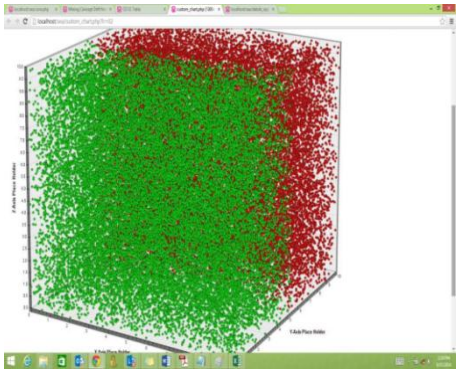

Figure 3 New Class Labels for Learning Rate 0.2

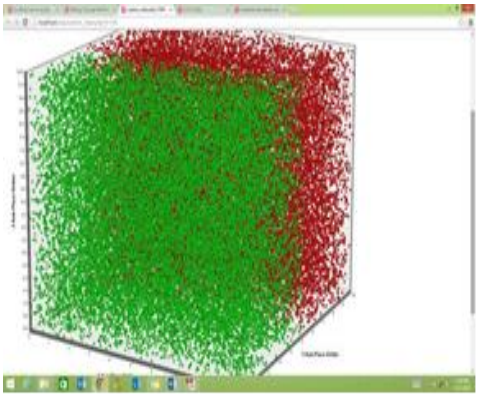

Figure4:New Class Labels for Learning Rate 0.3

Table6: Analysis for Learning rate of 0.4

Learning Rate	0.6
TotalClass 0	26390
Total Class1	23610
Correct Class 0(TP)	19513
TP%	63.65
Error Class0	6877
Diff Class 0(ACT-CORR)(FP)	11146
FP%	36.35
Correct Class 1(TN)	12464
TN%	64.44
Error Class 1	11146
Diff Class 1(ACT-CORR)(FN)	6877
FN%	35.56

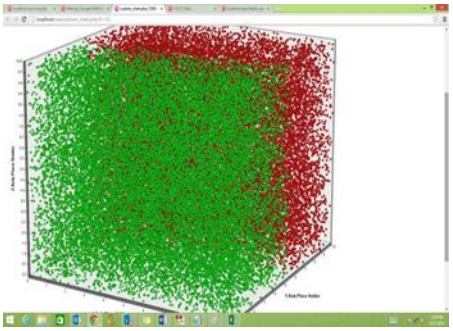

Figure 5 New Class Labels for Learning Rate 0.4

Table7: Analysis for Learning rate of 0.5

Learning Rate	0.5
TotalClass 0	25875
Total Class1	24125
Correct Class 0(TP)	19262
TP%	62.83
Error Class0	6613
Diff Class 0(ACT-CORR)(FP)	11397
FP%	37.17
Correct Class 1(TN)	12728
TN%	65.81
Error Class 1	11397
Diff Class 1(ACT-CORR)(FN)	6613
FN%	34.19

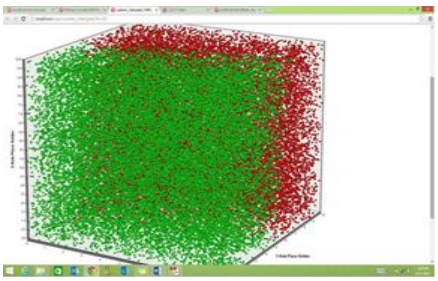

Figure 6 New Class Labels for Learning Rate 0.5.

Table8: Analysis for Learning rate of 0.6

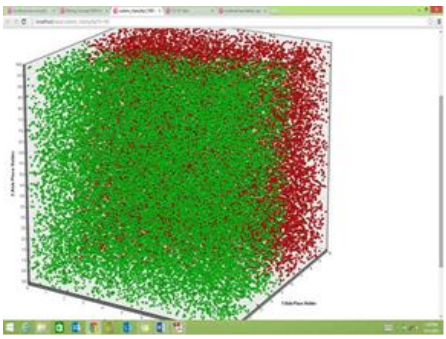

Figure 7: New Class Labels for Learning Rate 0.6

Table9: Analysis for learning rate of 0.7

Learning Rate	0.7
TotalClass 0	29354
Total Class1	20646
Correct Class 0(TP)	21433
TP%	69.91
Error Class0	7921
Diff Class 0(ACT-CORR)(FP)	9226
FP%	30.09
Correct Class 1(TN)	11420
TN%	59.05
Error Class 1	9226
Diff Class 1(ACT-CORR)(FN)	7921
FN%	40.95

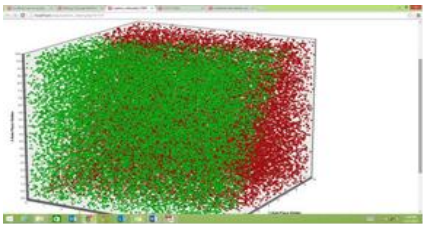

Figure8 : New Class Labels for Learning Rate 0.7

Table10: Analysis for Learning rate of 0.8

Learning Rate	0.8
TotalClass 0	29809
Total Class1	20191
Correct Class 0(TP)	21789
TP%	71.07
Error Class0	8020
Diff Class 0(ACT-CORR)(FP)	8870
FP%	28.93
Correct Class 1(TN)	11321
TN%	58.53
Error Class 1	8870
Diff Class 1(ACT-CORR)(FN)	8020
FN%	41.47

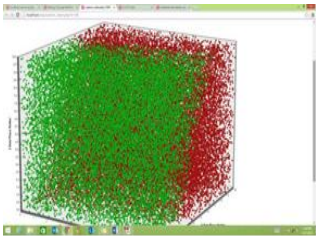

Figure 9 with New Class Labels for Learning Rate 0.8

Table11: Analysis for Learning rate of 0.9

Learning Rate	**0.9**
TotalClass 0	**29883**
Total Class1	**20117**
Correct Class 0(TP)	**22000**
TP%	**71.76**
Error Class0	**7883**
Diff Class 0(ACT-CORR)(FP)	**8659**
FP%	**28.24**
Correct Class 1(TN)	**11458**
TN%	**59.24**
Error Class 1	**8659**
Diff Class 1(ACT-CORR)(FN)	**7883**
FN%	**40.76**

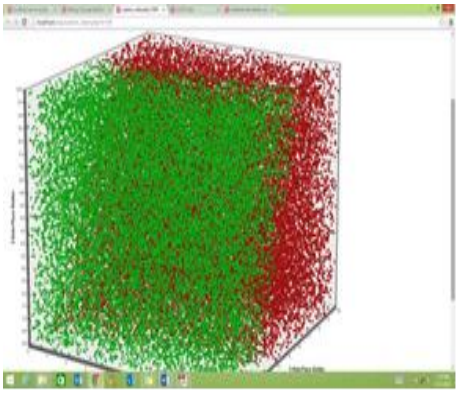

Figure 10: New Class Labels for Learning Rate 0.9

Table12: Analysis for Learning rate of 10

Learning Rate	**1.0**
TotalClass 0	**30134**
Total Class1	**19866**
Correct Class 0(TP)	**22536**
TP%	**73.51**
Error Class0	**7598**
Diff Class 0(ACT-CORR)(FP)	**8123**
FP%	**26.49**
Correct Class 1(TN)	**11743**
TN%	**60.72**
Error Class 1	**8123**
Diff Class 1(ACT-CORR)(FN)	**7598**
FN%	**39.28**

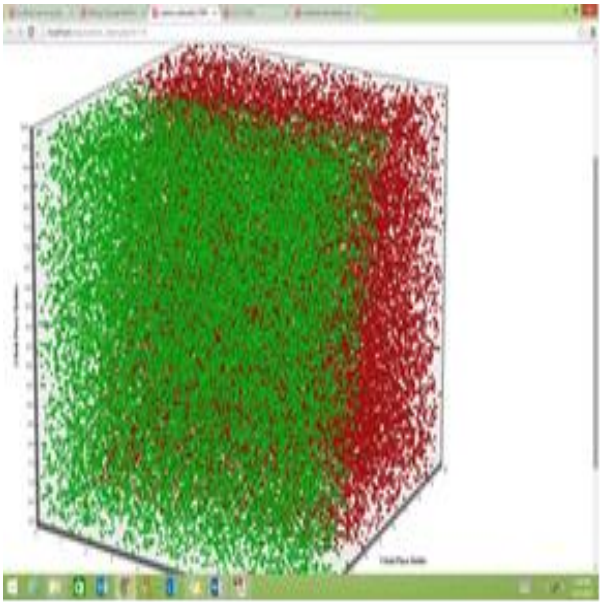

Figure11New Class Labels for Learning Rate 1.0.

5.11 Comparison of Results

Table13: Comparison of Results for different Learning Rates

Learning rate	Total Class0	Total Class1	Correctclass0	TP%	Error Class 0	Diff Class 0(ACT-CORR	FP%	Correct Class 1	TN%	Error Class1	Diff Class 1(ACT-CORR)	FN%
0.1	2458	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	22539	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 13, 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 the more drift detection & the maximum drift can be found at the learning rates of 0.7 & 0.9, as in the dataset [10] 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 & from the detected drift a variety of other conclusions on the data can be made.

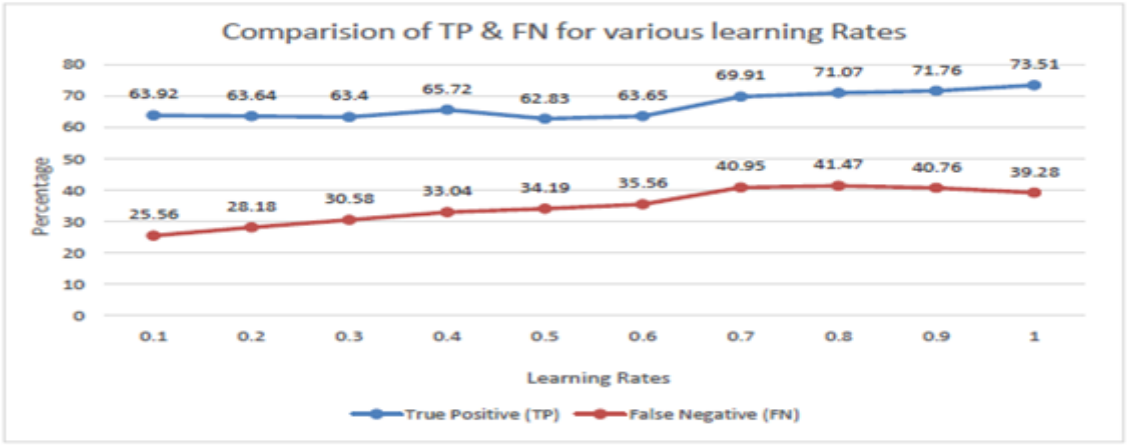

Figure 12: Comparison of TP & FN for Learning Rates 0.1 to 1.0.

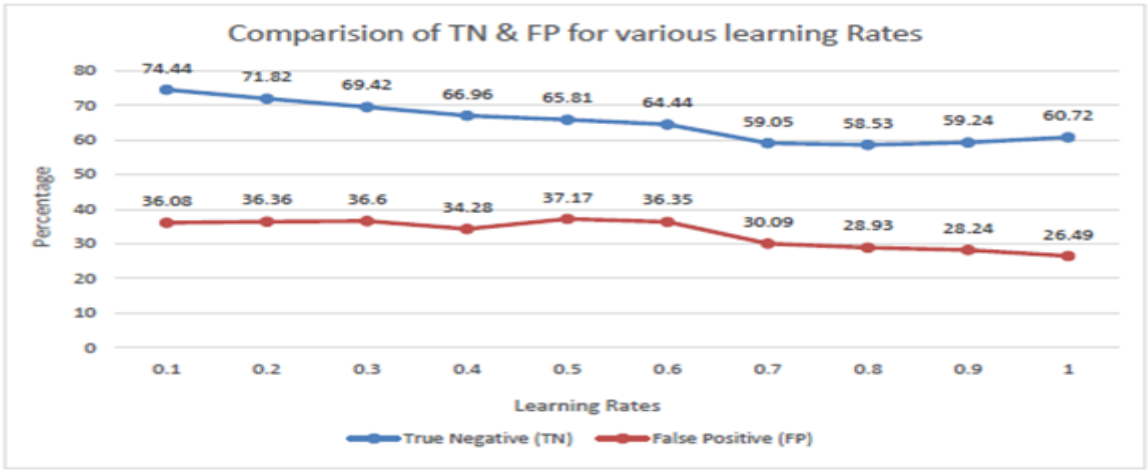

Figure 13: Comparison of TN & FP for Learning Rates 0.1 to 1.0

6. CONCLUSION AND FURTHER ENHANCEMENT

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 & help to find the drift in constantly changing Streams. The experimentation done was for the SEA Drift Set Database [10], which contains 50,000 records and 40% drift. As per the results, as the learning rate increased the Drift Detection in the Dataset too increased with25.6 % for learning rate of 0.1 to 39.2 % for the Learning Rate of 1.0. The most optimal solution was found for the learning rate of 0.7 & 0.9, these two learning rates can be said as the optimal learning rates for Drift Detection in this Dataset.Future work in this algorithm could be after finding the Drift in the Datasets making use of the Drift for Fraud detection or for other areas of Drift applications like Spam Detection.The current algorithm only works for numeric attribute values of the dataset. It can be enhanced for making it work for the Non-numeric value of the attributes and also for other areas of Concept Drift.

7. REFERENCES

[1] J. Gehrke, V. Ganti, R. Ramakrishnan, and W. Loh, ―BOAT— Optimistic Decision Tree onstruction,‖ Proc. *ACM SIGMOD Int'l Conf. Management of Data* (SIGMOD '99), 1999.

[2] 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.

[3] Jordan, Michael I.; Bishop, Christopher M. (2004). "Neural Networks‖. In Allen B. Tucker. *Computer Science Handbook*, Second Edition (Section VII: Intelligent Systems). Boca Raton, FL: Chapman & Hall/CRC Press LLC.[4] G. Widmer and M. Kubat, ―Learning in the presence of concept drift and hiddencontexts‖. *Machine Learning*, vol. 23, no. 1, pp. 69-101, 1996.

[4] W. N. Street and Y. Kim, ―A streaming ensemble algorithm (sea) for large-scale classification.‖ *ACM Press*, 2001, pp. 377– 382.

[5] J. Z. Kolter and M. A. Maloof, ―Using additive expert ensembles to cope with concept drift,‖ in *ICML*, 2005, pp. 449–456.

[6] GuénaëlCabanes and YounèsBennani, ―Change detection in data streams through unsupervised learning‖, *WCCI 2012 IEEE World Congress on Computational Intelligence*, 2012.

[7] J.Z.Kolter and M.A.Maloof,"Using additive expert ensembles to cope with concept drift ", in ICML,2005,pp,449-456.

[8] T.Kohonen "Self –Organizing Maps ".Berlin: Springer-Verlag,2001.

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

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

[11] B. Silverman, ―Using kernel density estimates to investigate multimodality,‖ *Journal of the Royal Statistical Society*, Series B, vol. 43,pp. 97–99, 1981.

[12] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, ―A framework for clustering evolving data streams,‖ in *Very Large Data Base*, 2003, pp. 81–92.

[13] Jordan,Michael I.;Bishop,Christopher M.(2004). "Neural Networks". In Allen B.Tucker. Computer science Handbook,Second Edition (Section VII:Inrelligent Systems).Boca Raton,FL:Chapman & Hall /CRC Press LLC.