Empirical estimation of various data stream mining methods

Ritesh Srivastava*

Computer Science and Engineering Department,
Galgotias College of Engineering and Technology,
Knowledge Park II, Greater Noida,
Uttar Pradesh 201310, India
Email: ritesh21july@gmail.com
*Corresponding author

Veena Mittal

Computer Science and Engineering Department,
Faculty of Engineering and Technology,
MRIIRS, Faridabad, India
Email: veena.mittal06@gmail.com

Abstract: Online learning is done in order to work on dynamic environments in which the concept tends to change with time and the accuracy of classifiers decreases. The current and previous research is done in static environments, but there is a need of a real time data streaming due to the potentially larger number of applications available in the scientific and business domains. There are several methods used in learning in the presence of dynamic environments like single classifier methods such as batch and incremental learning approaches, classification methods with explicit drift detection method, windowing techniques and ensemble approaches. This paper, investigates these approaches for determining the best suitable method among them. We utilised light emitting diode (LED) data generator for evaluating the performance of the methods.

Keywords: concept drifts; online learning; data stream mining; machine learning; classification; drift detection methods.

Reference to this paper should be made as follows: Srivastava, R. and Mittal, V. (2021) 'Empirical estimation of various data stream mining methods', *Int. J. Computational Intelligence Studies*, Vol. 10, No. 1, pp.13–26.

Biographical notes: Ritesh Srivastava has obtained his PhD in Computer Engineering and Master of Technology in Information System from University of Delhi, New Delhi, India. He is working as an Associate Professor in Computer Science and Engineering Department of GCET, Greater Noida, India. His research areas include machine learning (ML), data stream mining, text mining, information retrieval (IR) and natural language processing (NLP). He is an active member of various professional bodies and SIGs like IEEE, ACM, ACM-SIGKDD, Big Data Community, IEEE Computer Society and CSI-India. He is serving as reviewer of various journal and conferences like Springer, IEEE, AAAI, etc.

Veena Mittal holds a Master in Computer Science and Engineering from Banasthali Vidyapith, Rajasthan, India. She is currently working as an Assistant Professor in Computer Science and Engineering Department in FET of MRIIRS, Faridabad, India. Her research areas include machine learning, data stream mining and big data.

1 Introduction

In the era of information technology, the data can be gathered and shared at anytime and anywhere in the world. It is implemented in many application areas like social networking, credit applications, fraud detection, manufacturing, medicine, banking, health care, research, stock market, entertainment, telecommunications, retail, finance banks and credit applications, etc. The machines learning which is a field of computer science gives the ability to machines to learn without being programmed for it and therefore it can be used to make predictions based on the learning of algorithms. A huge amount of data gathered from the storage in computer systems is presented in the form of datasets on which these learning algorithms are implemented. The datasets may contain many useful and interesting hidden patterns which can be extracted as knowledge; therefore, the knowledge discovery methods are of great importance in data mining. These methods provide understandable patterns from very large data sets. The data mining techniques includes association rules, clustering and classification techniques. An enormous amount of data is available to decision makers that are not only large in size but also having a large variety and velocity which is difficult to handle using traditional methods of data mining. But there is a need to handle such data in order to extract knowledge out of them because such type of data is continuously increasing due to social area networking, we can say that today is an era of big data. The processing of such data is very challenging as the data is sourced from multiple sources and also it is having very complex and evolutionary relationship. The example of Big data is presidential debate between president Barack Obama and Governor Mitt Romedy triggered more than 10 billion tweets in just two hours. It is how fast the rate on which data is expanding. There are many more example available in these tweets, the moments that generated discussions of public interests related to Medicare and vouchers. It provides new means to judge the public opinion and generates feedback in real time which is more valuable. The speed and size of datasets make them impossible to store permanently in memory for which we have to employ scheme of forgetting mechanism. It is very challenging to deal with such data and it opens the door in research for mining such class of data as current research is dedicated to only static environments.

1.1 Data stream

It is a sequence of instances, having very fast speed of arrival which is impossible to store in a memory permanently. It is very large in size, due to which, most of the data mining approaches fail to process (Mittal and Kashyap, 2015, 2016; Nishida et al., 2005; Srivastava and Bhatia, 2017a, 2017b, 2017c). The constraints imposed by data streams are as follows:

1. The information is stored in memory in the form of small summaries to carry out computations and rest of the information is discarded as it is impossible to store each and every part of data from data streams. It imposes the constraint for limited usage of memory.
2. The elements from data streams should be processed in real time due to their fast arrival speed. Therefore, online learning is a need for data stream mining. It imposes the time constraint in which processing is to be done.
3. The distribution, which is generated over data streams tend to change with time, which may be irrelevant and proves harmful for current memory. It imposes the constraint to study the learning of evolving data streams rather than static learning algorithms.

1.2 Date stream mining

1.2.1 Online learning

The online learning is also termed as incremental learning and it focuses to process the incoming data from data streams sequentially and also takes care about the accuracy of the trained classifier in such a manner, as if, it has been trained on the whole data set. The classifier is updated in order to accommodate new training point (Nishida et al., 2005). An online classifier should possess the following qualities:

1. *Incremental*: The algorithm should be able to read blocks of data at a time, as the whole data is not available at the beginning.
2. *Single pass*: The data is available for only one time to the algorithm
3. *Limited time and memory*: The arrival speed of data is very high and it cannot be stored permanently in the memory so it should be processed within limited time and memory.
4. *Any-time learning*: The algorithm should give the best possible answer even when stopped in between without its conclusion.
5. *Forgetting mechanisms*: A classifier in data stream should be able to respond to the evolving data. It can be achieved by forgetting of out-dated data and it is difficult to select the data range which is to be remembered. The solution of this problem is to forget the training objects at a constant rate and it can be achieved by using a window having latest examples and used to train the classifier. The size of window is having trade-off between stability and flexibility. The small size window will adapt to the changes quickly but accuracy can be low due to insufficiency of data in the window. The large window size provides stability but may not be able to adapt to the changes quickly. A decay function is used to differentiate the impact of data points. This approach tries to give better results in order to maintain the balance between its accuracy and flexibility.

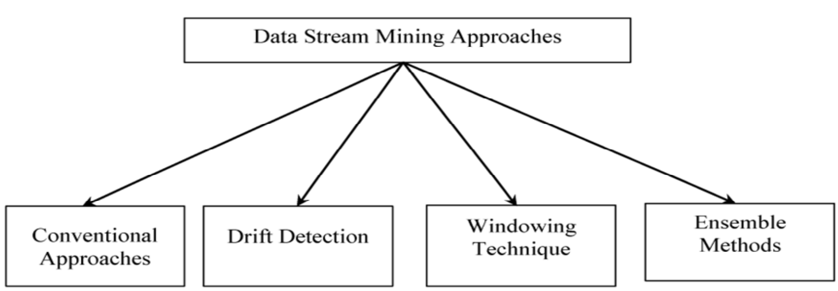
In the era of information technology, the data can be gathered and shared at anytime and anywhere in the world. It is implemented in many application areas like social networking, credit applications, fraud detection, manufacturing, medicine, banking, health care, research, stock market, entertainment, telecommunications, retail, finance banks and credit applications, etc.

The machines learning which is a field of computer science gives the ability to machines to learn without being programmed for it and therefore it can be used to make predictions based on the learning of algorithms. A huge amount of data gathered from the storage in computer systems is presented in the form of datasets on which these learning algorithms are implemented. The datasets may contain many useful and interesting hidden patterns which can be extracted as knowledge, therefore the knowledge discovery methods are of great importance in data mining. These methods provide understandable patterns from very large data sets. The data mining techniques includes association rules, clustering and classification techniques. An enormous amount of data is available to decision makers that are not only large in size but also having a large variety and velocity which is difficult to handle using traditional methods of data mining. But there is a need to handle such data in order to extract knowledge out of them because such type of data is continuously increasing due to social area networking, we can say that today is an era of big data. The processing of such data is very challenging as the data is sourced from multiple sources and also it is having very complex and evolutionary relationship. The example of Bigdata is presidential debate between president Barack Obama and Governor Mitt Romedy triggered more than 10 billion tweets in just two hours. It is how fast the rate on which data is expanding. There are many more example available in these tweets, the moments that generated discussions of public interests related to Medicare and vouchers. It provides new means to judge the public opinion and generates feedback in real time which is more valuable. The speed and size of datasets make them impossible to store permanently in memory for which we have to employ scheme of forgetting mechanism. It is very challenging to deal with such data and it opens the door in research for mining such class of data as current research is dedicated to only static environments.

1.3 Data stream mining algorithms

The four most commonly used methods for data stream mining which also includes the single classifier conventional approaches are given in the Figure 1. The rest three methods are listed and explained below:

Figure 1 Data stream mining approaches



1 *Drift detection methods*: This method is based on the fact that in each iteration, the decision class is predicted by using the online classifier. The number of classification errors is predicted using the random variable from Bernoulli distribution. The standard deviation is calculated using the probability of false prediction.

2 *Windowing techniques*: The sliding windows are used in dealing with time changing data by limiting the size of size of window from which the old concept is eliminated. It should have the property of any time learning and able to provide best answer at any moment in the processing of the input stream. The forgetting process is the key element of windowing techniques. The oldest concept that is not fitting in the window is discarded but includes the most recent examples.

3 *Ensemble methods*: The algorithm of ensemble is a set of components in which decision is formed by the aggregation of voting rule as given in Figure 2 (Srivastava and Bhatia, 2016) and generally computed using equation (1). A general ensemble method is given in Algorithm 1. The decision of ensemble is more accurate as compared to single classifier. The diversity in the components of ensembles can be made based upon the three parameters:

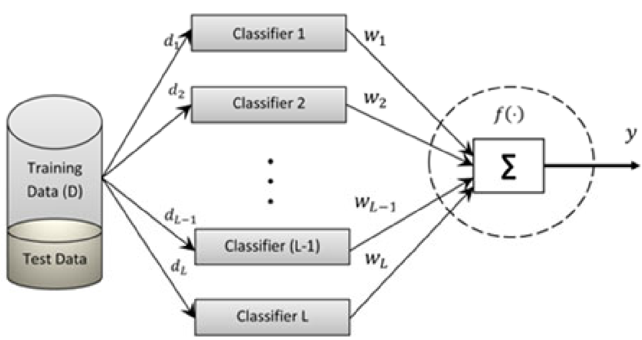
   a attributes used by the component

   b training data used for training of components

   c base learner from which they are created.

$$y_t \sum_j w_j \times Classifier_{ji} \text{ where } w_j \geq 0, \sum w_j = 1 \tag{1}$$

Figure 2 Voting rule in ensemble method (see online version for colours)



Algorithm 1 Ensemble method of learning

Input: $\mathcal{S}$ Set representing ensemble of classifier

n: No. of classifiers present in the ensemble $\mathcal{S}$

Output: $\mathcal{E}$ ensemble of the classifiers

$\mathcal{E} \leftarrow k$ *classifiers*

for all classifiers C_i in ensemble $\mathcal{E}$ do

Assign a weight to each example in $\mathcal{S}$ to create weight distributed D_m

Build/update C_i **with** $\mathcal{S}$ modified by weight distributed D_m

1.4 Bagging and boosting

Bagging and boosting are two commonly used methods for constructing ensembles; the remaining part of this section presents brief discussion on these two methods.

- *Bagging* (Breiman, 1996): It is a voting method in which base learners are trained over different training data as shown in Figure 3. It generates slightly different samples from a given sample of size N. The samples can be similar as they are taken from the same sample but can be different due to random replacement of samples. The learning algorithm is called unstable algorithm when small changes made in the training set and it gives more impact in the generated learner. In bootstrap aggregating, it generates L (L < N) base learners using this unstable algorithm, averaging is done during testing. This method is applicable to regression also but instead of average, median is used at the combining the predictions.

Figure 3 Bagging method (see online version for colours)

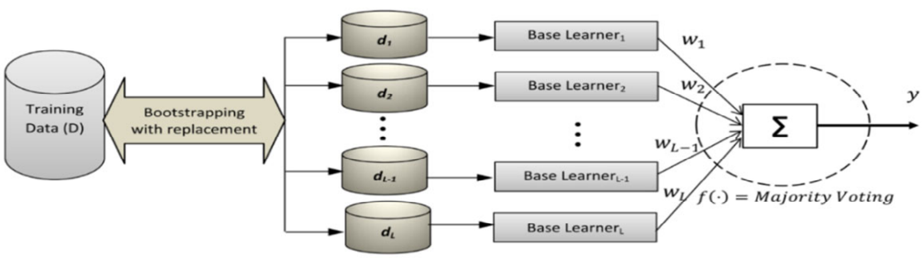
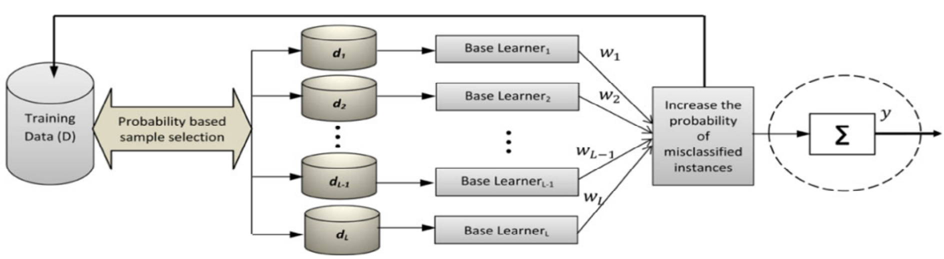


Figure 4 Probability based sample selection (see online version for colours)



- *Bootstrap with replacement (boosting)* (Freund and Schapire, 1997): In this technique, the next learner is trained on the mistakes (misclassification) of the previous learners in order to generate complimentary base learners as shown in Figure 4. It combines three weak learners and generates one strong learner, which is having a small error probability. The disadvantage of this approach is that it requires a large training set which is divided into three partitions randomly. The first classifier is trained on first partition of dataset. It takes all the instances which first classifier misclassifies for the second partition and many instances which are correctly classified will together form the dataset for second partition on the next classifier is trained. The third partition data is exposed to both the previous two classifiers and all the misclassified data using both of them will become the data set for third classifier.

In the testing phase, an instance is given to first two classifiers and if misclassified then the classification from the third one is considered as output. In case of small size of datasets, the second and third classifiers will be trained only a subset of dataset which are misclassified by first classifier.

2 Related work

The literature review is divided into four sections. Section 2.1 deals with the traditional single classifier methods. Section 2.2 presents the work that utilised explicit drift detection method along with some base classifiers. Section 2.3 in the literature survey deals with the mining of data streams using windowing techniques. Section 2.4 discusses the classification in data stream mining using the ensemble approach.

2.1 Single classifier approach

The classical single classifier methods like naïve Bayesian (NB) (Langley et al., 1992) and support vector machine (SVM) (Cortes and Vapnik, 1995) are among the commonly used classification methods for static environment. The NB method is a probabilistic classifier based on conditional probability as given in equation (2), where C is the set of class labels, x is an instance. The NB assumes that the features are independent from each other.

$$P\left(C/x\right) = \frac{P(C)P\left(x/C\right)}{P(x)} \tag{2}$$

SVM is also very popular classification algorithm. SVM is a discriminative classification model that is defined by separating hyper plane. SVMs are basically linear classifiers that learn decision boundary as given in equation (3), where is weight vector and b is threshold. SVM try to select the hyper-plan with maximum margin.

$$f(x) = sign\left(w^T x + b\right) = \begin{cases} +1 \text{ if } w^T x + b > 0 \\ -1 \text{ otherwise} \end{cases} \tag{3}$$

2.2 Methods with explicit drift detection method

Gama et al. (2004) proposed a method called drift detection method (DDM) as described in Algorithm 2. It is assumed that error rate of online classifier decreases with advancement of time if the target concept is stationary. But error rate will increase significantly in case of changing concept. There are two thresholds in DDM, one for warning level and another one is for drift level. In warning level, it stores the examples in a short-term memory for the rebuilding of online classifier. On the other hand, upon reaching of drift level, DDM reinitialises all the variables. The advantage of this approach is that it works well for in the detection of sudden and significant changes. The disadvantage of this approach is that it is not capable in detecting small and gradual changes.

Algorithm 2 Drift detection method

Input data stream, classifier,
Output Points for rebuilding of new classifiers

Initialise n, r, $p_{\min}$, $S_{\min} \leftarrow 0$, $\boldsymbol{B} \leftarrow \Phi$, p_t **misclassification probability**, s_t:
While (Data streaming)
    {
    for all instances $\in$ data stream
    {
        n++;
        if (classification)
            {
                r++;
            }
            $p_t \leftarrow r / n,\ s_t \leftarrow \sqrt{\dfrac{p_t(1-p_t)}{n}}$
        if ($p_t + s_t < p_{\min} + S_{\min}$)
            {
                $p_{\min} \leftarrow p_t$; $s_{\min} \leftarrow s_t$
            }
            Train classifier on (x_t, y_t)
                {
                if ($n \geq 30$)
                    {
                        if $(p_t + s_t < p_{\min} + 3s_{\min})$
                            {
                                rebuilt the classifier
                                initialise n, r, $p_{\min}$, $S_{\min} \leftarrow 0$, $\boldsymbol{B} \leftarrow \Phi$
                            }
                        else if ($p_t + s_t < p_{\min} + 2S_{\min}$)
                            {
                            add (x_t, y_t) **to** B
                        else
                            reinitialise B;
                    }
                }
    }

Baena-Garcıa et al. (2006) proposed a method called early drift detection method (EDDM). It is an improved form of DDM to detect small and gradual changes. It considers the distance between the occurrences of two classification errors. When this distance decreases significantly, then it assumes the occurrence of drift. It uses the measure of average distance calculation between two errors and this measure is more sensitive towards gradual changes as compared to DDM. Like DDM, it also stores examples in short term memory upon reaching of warning level and rebuilds an online classifier upon reaching the pre-specified drift level.

Nishida and Yamauchi (2007) proposed an approach called STEPD. This method monitors the two predictive accuracies called recent accuracy and overall accuracy for the online classifier. It is based on the assumption that there is statistical significance between these two accuracies after the change in the concept of target class. Furthermore, the statistical test of equal proportions is used to compare these two accuracies. Like DDM and EDDM it is also having two levels: warning level and drift level and therefore examples are stored in short term memory till the occurrence of warning condition and reinitialises all the variables of an online classifier upon reaching the drift level in the same manner as that of DDM and EDDM. This method works better in detection of gradual changes as compared to EDDM and much faster than DDM in case of sudden changes.

2.3 Window based data stream mining techniques

Cohen and Strauss (2003) suggested the use of decay functions for the implementation of forgetting process and making it more dynamic. This technique is called as weighted windows in which lesser weights are assigned to older process and therefore will be treated as of less importance by base classifier. This method analysed the use of decay functions of different types to calculation of data stream aggregates.

Bifet and Gavalda (2006, 2007) proposed the concept of sliding window which is adaptive and the algorithm is called as ADWIN. It is suitable for the sudden changes in the data streams. The idea behind this algorithm is that it keeps a sliding window having the most recently read examples and when two sub-windows are having distinct enough averages then there can be a change between corresponding values and detection of sudden changes is considered, also, the portion of window which is older is dropped. Oza-ADWIN has been proposed in Bifet et al. (2009), that utilised ADWIN triggers. Žliobaitė (2010) proposed a family of algorithms and it is called as FISH. The family consists of FISH1, FISH2 and FISH3. The family use the similarities of time and space between examples to create a window dynamically. Klinkenberg and Joachims (2000) and Tsymbal et al. (2008) proposed two windowing methods that are incorporated by FISH2 and FISH3 in order to implement a window of variable sample size.

2.4 Ensemble based method for data stream mining

Street and Kim (2001) proposed an ensemble algorithm called streaming ensemble algorithm. It changes the structure of an algorithm so that it can react to changes. The idea of this algorithm is the replacement of weakest expert based on accuracy and diversity. The accuracy is used because it is the most prime factor, that the most recent examples should be correctly classified in order to detect and adapt to changes. The diversity uses in bagging and boosting which are the techniques used to give the boost to ensembles. Wang et al. (2003) proposed an algorithm called accuracy weighted ensemble (AWE) which is used for restructuring of ensembles. A new classifier is trained on the most recent data chunks, which is then used to evaluate the existing ensemble members in order to select the best component classifiers.

Pfahringer et al. (2007) proposed an ensemble method which uses Hoeffding trees and the method is called as Hoeffding option tree (HOT). It is similar to the method named option decision trees which is proposed by Kohavi and Kunz (1997). In option decision tree, the decision path splits the tree into several sub-trees and also the training example updates a set of option nodes instead of a single leaf. It combines the prediction of all applicable leaves into one result while making a decision. The HOT works as a set of weighted classifiers in the compact structure provided by it. Like Hoeffding trees, HOT is also built in an incremental fashion.

Bifet (2009) also proposed an ensemble method called adaptive-size Hoeffding tree bagging (ASHT) and it is strictly designed for Hoeffding trees. It provides the diversity in the components of ensembles by using different sizes trees because the smaller trees are more adaptable to changes. It also provides the forgetting mechanism that is very useful in data stream mining. Oza and Russell (2001) proposed an online version of bagging and boosting techniques in which component classifiers combine their decision using a simple majority vote and these classifiers are in incremental fashion, therefore, they are also called as incremental learners.

Brzeziński and Stefanowski (2011)proposed an algorithm called accuracy updated ensemble (AUE). This algorithm is designed to react equally to all types of changes. The existing classifiers work well for any single type of change. Some of them are good for detection of gradual changes while other are good for sudden changes, most of them specialise for one type of change. In AUE, the accuracy based weighted mechanism from block-based ensembles are combined with the incremental nature of Hoeffding trees in order to incorporate all types of changes.

3 Experimental setup and data collection

We conducted experiments on massive online analysis (MOA) tool, which is an extension of WEKA. We performed our experiments on Intel dual cure i3 1.8 GHz processor with 3MB L3 Cache and 12 GB of RAM. We selected following combinations of various approaches with one popular instance of each approaches as shown in Table 1.

Table 1 Approaches of data stream mining with their corresponding instances

Data stream mining approaches	*Algorithm*
Conventional approaches	Naïve Bayesian
Drift detection methods (DDM)	DDM with HT
Windowing techniques	Oza_ADWIN
Ensemble	AWE

3.1 Data collection

We used light emitting diode (LED) data generator implemented in MOA to generate LED data stream. The LED data stream generators generates data stream with 24 attributes. The LED represents variations of seven segments and contains abrupt drifts. We generated 1million (M) instances and introduced one drift after 100,000th (1Lakh) instance with 20% noise.

4 Result analysis

This section presents the results and analysis of experiments that we conducted on LED data stream generated by using LED data generator on our selected data stream classification methods. We utilised prequential accuracy to measure the performance of each classification approach and also considered the learning time to compute the time consumed by each algorithm for learning. To evaluate the average accuracy, we selected window size of 500 instances. We depicted the results of performed experimental evaluations by Table 2 and Figure 5 respectively, the Table 2 summarised the average accuracy and learning time consumed by all selected combinations and the Figure 5 depicts the graphical representations for observing the run time analysis of all selected methods. The X-axis of the graphs in Figure 5, shows the number of instances and the Y-axis shows the average accuracy percentage.

Figure 5 Percent accuracy of classifiers on data stream generated by LED generated with 1 million instances and a drift introduced at 100,000th instance (a) Naïve Bayesian (NB) (b) DDM with NB (c) AWE (d) Oza_ADWIN (see online version for colours)

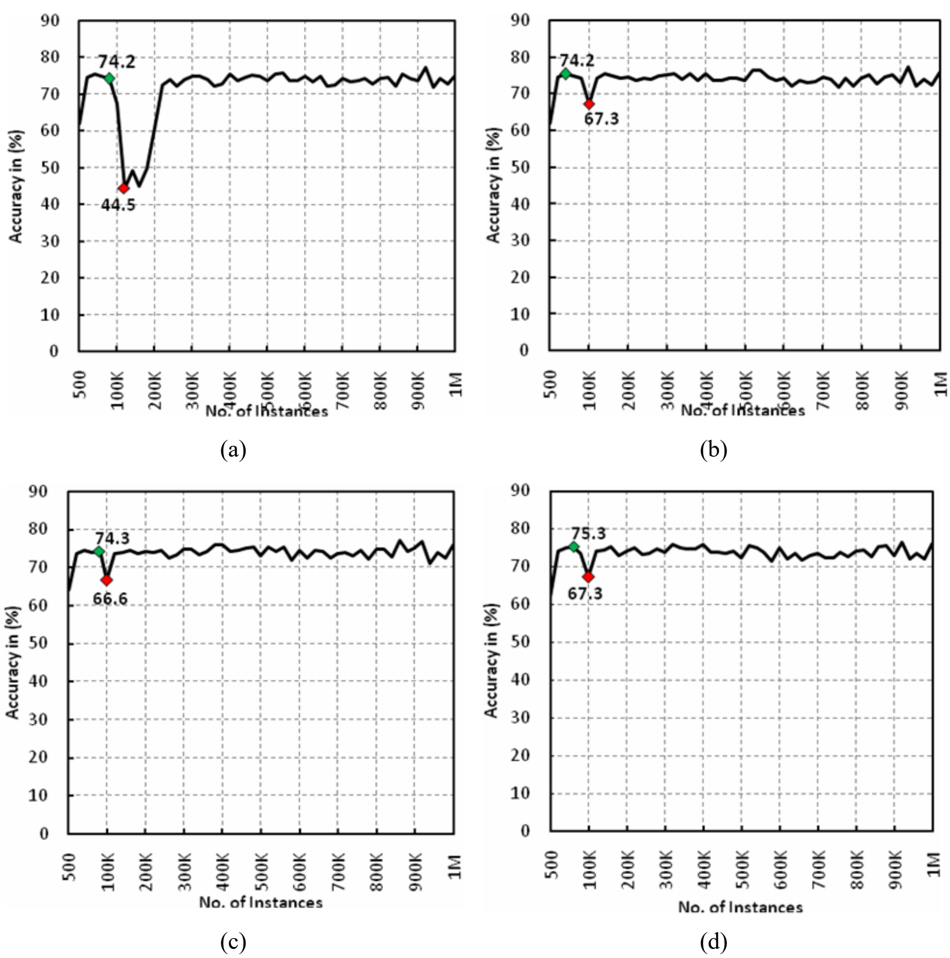


Table 2 Prequential accuracy percentage

Data stream classification method	*Accuracy in percentage*	*Learning time in seconds*
NB	71.38%	8.08
DDM	73.88%	12.83
AWE	74.03%	180.06
Oza-ADWIN	73.84%	115.23

From Table 2, it can be observed that the single classifier Naive Bayesian (NB) approach is the worst performer with an average accuracy of 71.38% and the ensemble based method AWE is the best classifier with an average accuracy of 74.38%, however the AWE consumed very large learning time of 180.06 seconds as compared to NB which takes 8.08 seconds only, the rest two approaches, that is, the drift detection based method DDM with NB and Oza-ADWIN shown average accuracy of 73.88% and 73.84% and consumed learning time of 12.83 seconds and 115.23 seconds respectively.

From all graphs in Figure 5, it can be easily observed that a drift is introduced at position of the 100,000th instance of the data stream generated by LED generator. The graphs clearly depict that the average accuracy of all classifiers drops after 100,000th instance because of the introduced drift.

Figure 5(a) shows the run time average accuracy of single classifier-based method Naïve Bayesian (NB). From the graph of the Figure 5(a), it can be observed that in the window of 100000th instance, the accuracy of the classifier drops drastically with 29.7% from 74.2% to 44.5%.

The graph in Figure 5(b) depicts the run time average accuracy of drift detection-based method (DDM) with NB, the graph shows that on the arrival of drift the average accuracy of the classifier drops from 74.2% to 67.3%. The run-time average accuracy of ensemble-based method AWE is depicted by the graph in Figure 5(c), the graph shows that on the occurrence of drift, the accuracy of the classifier drops from 74.3% to 66.6%. Similarly, from the graph of Figure 5(d), it can be observed that in the window-based method ADWIN the accuracy of the classifier drops from 75.3% to 67.3%.

From the graphs of Figure 5, it can be easily observed that the single classifier-based method NB is the worst performer in reacting in the presence of drift and the other three methods are extremely better than single classifiers approach. On comparing the rest three approaches with each other, we observed that the AWE is slightly better than the rest two methods.

5 Conclusions

The real-world environment is full of concept drifting data stream generators therefore, it is very important to have classification methods that should be able to perform equally well in the presence of concept drifting data stream also as they perform in static environment. To achieve consistent performance in dynamic environment many approaches like drift detection-based methods, ensemble-based methods and window-based methods are proposed. In this work, we evaluated the performance of these methods and also compared the performance of all of them with each other as well as with the classical single classifier method. We observed that drift detection-based methods, ensemble-based methods and window-based methods outperforms over the classical single classifier method on LED data stream with concept drift. We also observed that the AWE performs slightly better than rest methods; however, it takes very large learning time as compared to other approaches of classification we have selected for comparison.

References

Baena-Garcıa, M., del Campo-Ávila, J., Fidalgo, R., Bifet, A., Gavalda, R. and Morales-Bueno, R. (2006) 'Early drift detection method', in *Fourth International Workshop on Knowledge Discovery from Data Streams*, September, Vol. 6, pp.77–86.

Bifet, A. (2009a) 'Adaptive learning and mining for data streams and frequent patterns', *ACM SIGKDD Explorations Newsletter*, Vol. 11, No. 1, pp.55–56.

Bifet, A. and Gavalda, R. (2006) 'Kalman filters and adaptive windows for learning in data streams', in *International Conference on Discovery Science*, Springer, Berlin, Heidelberg, October, pp.29–40.

Bifet, A. and Gavalda, R. (2007) 'Learning from time-changing data with adaptive windowing', in *Proceedings of the 2007 SIAM International Conference on Data Mining*, Society for Industrial and Applied Mathematics, April, pp.443–448.

Bifet, A., Holmes, G., Pfahringer, B., Kirkby, R. and Gavaldà, R. (2009b) 'New ensemble methods for evolving data streams', in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, pp.139–148.

Breiman, L. (1996) 'Bagging predictors', *Machine Learning*, Vol. 24, No. 2, pp.123–140.

Brzeziński, D. and Stefanowski, J. (2011) 'Accuracy updated ensemble for data streams with concept drift', in *International Conference on Hybrid Artificial Intelligence Systems*, Springer, Berlin, Heidelberg, pp.155–163.

Cohen, E. and Strauss, M. (2003) 'Maintaining time-decaying stream aggregates', in *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, ACM, June, pp.223–233.

Cortes, C. and Vapnik, V. (1995) 'Support-vector networks', *Machine Learning*, Vol. 20, No. 3, pp.273–297.

Freund, Y. and Schapire, R.E. (1997) 'A decision-theoretic generalization of on-line learning and an application to boosting', *Journal of Computer and System Sciences*, Vol. 55, No. 1, pp.119–139.

Gama, J., Medas, P., Castillo, G. and Rodrigues, P. (2004) 'Learning with drift detection. In Brazilian symposium on artificial intelligence, Springer, Berlin, Heidelberg, September pp.286–295.

Klinkenberg, R. and Joachims, T. (2000) 'Detecting concept drift with support vector machines', in *ICML*, June, pp.487–494.

Kohavi, R. and Kunz, C. (1997) 'Option decision trees with majority votes', in *ICML*, July, Vol. 97, pp.161–169.

Langley, P., Iba, W. and Thompson, K. (1992) 'An analysis of Bayesian classifiers', in AAAI, July Vol. 90, pp.223–228.

Mittal, V. and Kashyap, I. (2015) 'Online methods of learning in occurrence of concept drift', *International Journal of Computer Applications*, Vol. 117, No. 13, pp.18–22.

Mittal, V. and Kashyap, I. (2016) 'Empirical study of impact of various concept drifts in data stream mining methods', *International Journal of Intelligent Systems and Applications*, Vol. 8, No. 12, pp.65–72.

Nishida, K. and Yamauchi, K. (2007) 'Detecting concept drift using statistical testing', in *International Conference on Discovery Science*, Springer, Berlin, Heidelberg, October, pp.264–269.

Nishida, K., Yamauchi, K. and Omori, T. (2005) 'ACE: adaptive classifiers-ensemble system for concept-drifting environments', in *International Workshop on Multiple Classifier Systems*, Springer, Berlin, Heidelberg, June, pp.176–185.

Oza, N.C. and Russell, S. (2001) *Online Ensemble Learning*, University of California, Berkeley.

Pfahringer, B., Holmes, G. and Kirkby, R. (2007) 'New options for Hoeffding trees', in *Australasian Joint Conference on Artificial Intelligence*, Springer, Berlin, Heidelberg, December, pp.90–99.

Srivastava, R. and Bhatia, M. (2017a) 'Challenges with sentiment analysis of on-line micro-texts', *International Journal of Intelligent Systems and Applications*, Vol. 9, No. 7, pp.31–40.

Srivastava, R. and Bhatia, M. (2017b) 'Offline vs. online sentiment analysis: issues with sentiment analysis of online micro-texts', *International Journal of Information Retrieval Research (IJIRR)*, Vol. 7, No. 4, pp.1–18.

Srivastava, R. and Bhatia, M. (2017c) 'Real-time unspecified major sub-events detection in the twitter data stream that cause the change in the sentiment score of the targeted event', *International Journal of Information Technology and Web Engineering (IJITWE)*, Vol. 12, No. 4, pp.1–21.

Srivastava, R. and Bhatia, M.P.S. (2016) 'Ensemble methods for sentiment analysis of on-line micro-texts', in *2016 International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*, IEEE, December, pp.1–6.

Street, W.N. and Kim, Y. (2001) 'A streaming ensemble algorithm (SEA) for large-scale classification', in *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, August, pp.377–382.

Tsymbal, A., Pechenizkiy, M., Cunningham, P. and Puuronen, S. (2008) 'Dynamic integration of classifiers for handling concept drift', *Information Fusion*, Vol. 9, No. 1, pp.56–68.

Wang, H., Fan, W., Yu, P.S. and Han, J. (2003) 'Mining concept-drifting data streams using ensemble classifiers', in *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, August pp.226–235.

Žliobaitė, I. (2010) *Adaptive Training Set Formation*, Doctoral dissertation, Vilnius University.