A Survey on Classification of Concept Drift with Stream Data

Shweta Vinayak Kadam
Department of Intelligent Systems
University of Passau
Passau, Germany
kadam02@gw.uni-passau.de

***Abstract*—Usually concept drift occurs in many applications of machine learning. Detecting a concept drift is the main challenge in a data stream because of the high speed and their large size sets which are not able to fit in main memory. Here we take a small look at types of changes in concept drift. This paper discusses about methods for detecting concept drift and focuses on the problems with existing approaches by adding STAGGER, FLORA family, Decision tree methods, meta-learning methods and CD algorithms. Furthermore, classifier ensembles for change detection are discussed.**

***Index Terms*—Concept drift, Classification, data stream, mining**

I. Introduction

Nowadays, the new hardware technology is allowing us to record the transactions and other information automatically at a high amount, and such data grows at a significant rate. This amount of data consumes a lot of main memory; to overcome this, we can use a linear scan so that it would be cost-efficient [1]. Data streams always appear in the timeline as they are seriously influenced by the time and characteristics which are subject to change. Because of such reason stream data should be updated continuously or periodically for current information so that the model can come up with the latest report [1]. Data stream mining deals with real-time data.

Classification of a data stream is an essential area of machine learning. It has been assumed that the data has a stationary distribution because of its traditional classification technique [2]. The application of data stream involves email fraud detection, text mining, etc. Current state-of-the-art techniques in machine learning and pattern recognition fall short to explain modern challenges faced during classification, this classification of a data stream is one of those challenges, where the data distribution changes over time [3]. The classification adjusts to the new distributions upon detection of changes and tries to retain the classifier automatically [3].

Mining of data stream has achieved much attention over the past few years; different strategies have been used in mining techniques to handle the high speed and large size of stream data [5]. The primary data mining algorithms have focused on clustering, classification, and different pattern analysis techniques; because of extensive streaming information detection of changes in a data stream is an essential process [5]. One of the challenges in data stream classification is concept drift.

In general, concepts are not stable they change according to time. Examples of this is weather prediction rules and customers preference. In this the basic and hidden data gets changed; to build a new model, old data conflicts with new data by using those changes. That's why the model should be updated. Such a problem is known as Concept Drift [6]. "Changes in the hidden context can induce more or less radical changes in the target concepts, producing what is generally known as concept drift in the literature (e.g., Schlimmer and Granger, 1986)." [7][8]. Currently, classification of a data stream with concept drift is an important issue in data stream mining area.

To keep the classification model up-to-date for the data stream with concept drift is a serious challenge. For this there are some parameters [14] which include the following:

- $Accuracy$: It is difficult to recognize which examples represent the old concepts; therefore their results should be removed from the model. However, the higher rate of accuracy model will decrease the accuracy of the up-to-date model. Having a lower rate will make the model less sensitive and prevent it from discovering transient patterns.
- $Efficiency$: Based on the divide-and-conquer manner the decision trees are constructed, and they are unstable. Learning efficiency can be severely compromised when there is a slight drift in the underlying concepts which may trigger significant changes.
- $Ease\ of\ Use$: To handle data streams with the concept of drift in an incremental way it is required to adapt to classification methods of decision trees. Since the state-of-the-art methods cannot be implemented directly, there is some limitation to the usability of this approach.

II. Organic Computing and Concept Drift

In today's world, systems are becoming autonomous with the introduction of artificial intelligence. "Organic computing is an elementary field of system engineering with the goal to make technical systems more "life-like" (organic) by presenting them with their abilities."[41],[42]

Nowadays, computerization of the environment is providing the number of applications to us, some of them have a problem with controllability, which means it is essential to create a new system which will be flexible, robust and trustworthy; to achieve these goals the system must have to act like more autonomously, i.e., they will have to be "life-like" (organic) [43]. Hence "organic computer" is a technical system which adapts the current condition of the environment.

Organic computing makes the use of some concepts such as self-* properties [42] namely: self-organization, self-configuration, self-integration, self-management these properties make an impact on system's behavior; remaining self-* properties are also impressive in organic computing such as self-healing, self-protection, self-stabilizing, self-improving, self-explaining [42]. The concept of self-* properties is, it allows the system to react to component failures or changing environment [44].

Concept drift occurs due to the hidden context in the model [7], in this the data changes over time, which leads to poor performance of the model. Methods for handling the concept drift are designed in such a way that it can detect the occurrence of concept drift easily or predict the presence of concept drift.

Some algorithm of concept drift allows the classifier to update the model by its own. This property is similar to self-configuration where parameters in the system are modified according to the higher-level user goal also the system improves its performance on its own when the new data is introduced in the system which represents the self-improving property [42].

III. Related work

There are numbers of approaches available in machine learning to handle the concept drift. Some reviews of approaches are listed here.

STAGGER [8] is the first concept drift handling system which was introduced in 1986 by Schlimmer and Granger; to recognize concept drift, STAGGER decreases its probability over time. FLORA [7] was launched in 1996. FLORA framework is a window adjustment method which deals with one instance at a time due to its limitation of the high speed of coming data.

Another approach is based on the decision tree method known as VFDT [31] which was introduced in 2000 and CVFDT [27] in 2001. VFDT is a primary extension of decision tree learning algorithm, which was proposed by Domingos and Hulten. Moreover, CVFDT is an extended version of VFDT which includes all benefits of VFDT. It uses a sliding window to handle the concept drift.

There are several approaches related to ensemble classifier. One of the most famous is SEA (streaming ensemble algorithm) [34], which was proposed by Street and Kim in 2001; they suggest that to handle the concept drift, the data should be divided into fixed size chunks and by using those chunks ensemble classifier can be built to handle the concept drift.

Wang et al. [14] have proposed another ensemble classifier known as AWE (Accuracy Weighted Ensemble). In this method, by using the training set, the classifiers are built. It always built the classifier according to their performance. Some instance weighting uses some learning algorithm to process weighted instances such as SVM (Support Vector Machine) which was introduced by Klinkenberg in 2004 [13]. Bifet proposed another sliding window algorithm known as ADWIN, which works more accurately with a sudden drift [11].

IV. Approaches for handling Concept drift

A. Single classification approach

Traditional learners are well-known classifiers, which are used in data mining to satisfy their stream mining requirements by using their qualities [10]. They have some characteristics of online learner and a forgetting mechanism. Some methods like Naive Bayes, Neural network and Decision tree rules are used. Windowing technique is an approach which deals with time changing data that involves sliding windows [10]. It limits the number of examples which are introduced to the learner. Windowing techniques include some methods like FISH, ADWIN, and weighted windows. Algorithms of drift detector allow to adapt any learner to evolve stream data. When concept drift is detected they alarm the base learner to update or rebuild the model. Upon using DDM and EDDM concept drifts are detected.

B. Sample-based approach

Sample selection and sample weights are the categories of sample-based approaches [9].

Most common methods which are dealing with concept drift are based on sample selection. It selects the sample which is related to the latest concept and disregards the samples of old content. This method keeps a fixed or size variable sliding window; the learning system uses the concepts which are learned from a sliding window to predict the class label reaching at the next moment [9]. The sliding window could be of a non-variable size; such sliding windows can be adjusted by using some methods such as FLORA2 [7] and ADWIN2 [11] etc. These algorithms monitor the changes in classification accuracy and detect the occurrence of concept drift, once it is detected it adjust the sliding window size by a degree of shift [9].

When a data stream is reached as data blocks the selection of blocks can be viewed as sample selection, samples which are selected in more than one block can be grouped by latest classification model which indicates that those samples in the block are closely related to latest target concept [9]. Since a large number of windows are used in detecting concepts, it is divided into Single-window detecting method and multi-window detecting method [9].

As time passes, the significance of the sample should be decreased gently. This is the main concept of sample weights. The importance of sample can be represented by weight, time t is assigned to the weight when the sample arrives [9]. Some learning algorithms are utilized to handle sample weights when the weights of all samples are settled in the training set.

C. Explicit detection method and Implicit detection method

Explicit detection method and implicit detection method are two different methods of concept drift detection. To design a fast and accurate detection algorithm with a low false alarm rate is the main task of the explicit detection method. Some features should be satisfied by the algorithm of explicit detection method such as it should be able to identify and reuse the repeated information. And the detection algorithm should easily coordinate with the classification learning algorithm. The algorithm should have great anti-noise performance because noise can make changes in the object concept [7]. So the algorithm has to be efficient enough to identify the difference between concept drift and noise. When the occurrence of concept drift is detected, it informs the classification model to take appropriate action such as updating the classifier and setting a window of current data by explicitly using some kind of concept drift detection method.

The implicit detection method is used by most of the ensemble classifiers [14], when concept drift occurs the weights of the classifier changes which means changes of the weights shows that concept drift has occurred [3]. To compare the similarities and differences between concepts of different time is a challenging task which every (explicit or implicit) concept drift detection method have to perform [13]. Currently, Concept drift detection method tracks concept drift from two aspects: reasons that may cause concept drifts or possible effects after concept drift [15]. Such detection techniques include the following:

1) Probability distribution: Data streams assume the common probability distribution function typically generates all the data which is processed, but in the case of the evolutionary data stream with concept drift, the probability distribution of data could change over the time [15]. Therefore by observing the change in the probability, distribution of the data can detect the occurrence of concept drift [15]. Furthermore [16],[17], [18], [19] prior probabilities (past data) will be forgotten if the new data will not fit in the old data distribution.

2) Feature relevance: The sample characteristics (attributions) are changed if there is an occurrence of concept drift; former suited characteristics may no longer be relevant [20]. Therefore by following or monitoring the relation between numerous characteristics one can determine whether the concept drift has occurred or not. The classification model could be trained for the latest data distribution by tracking the best combination of predictive features.

3) Model complexity: Few classification models are very delicate to change in the data distribution [9]. For example, an explosion of some rules in the rule-based classifier or surge in the number of support vectors in the support vector machine indicates the occurrence of concept drift [9].

4) Classification accuracy: Classification accuracy is the most used criteria by concept drift detection algorithm [9]. Classification algorithm (explicit and implicit) uses classification accuracy to check the appearance of concept drift. This group involves : Winnow variants [21], [22], AdaBoost variants [23], method based on random decision tree [24], accuracy-weighted ensembles[14], etc. Moreover, In some classification algorithm, classification accuracy is used as an indicator; where some indicator such as recall, precision, etc. or series of them come under this class [12].

5) Time stamp: The time stamp of single or block stamp can be taken as additional input attribute. It is helpful to check whether the concept drift occurs or not according to the rule with timestamp attribute. This technique is probably used in time-changing concepts such as CD3, CD4, and CD5, this series of an algorithm is described in [4], [27]. While constructing the decision tree for the model, it uses the timestamp as an additional input attribute. Regarding distributed data streams, there is no presence of time stamp attribute in any path of the decision tree and this condition of the time stamp is not suitable for other properties [9]. The time stamp appears in when the concept drift occurs. When some classification path consists the value of time stamp attribute, it describes that this classification path is showing previous or old time in a decision tree which means that the classification rule is too old or outdated [9]. For this reason, it can not be used for next time to classify the data.

V. Types of concept drift

By generalizing [6], [15], There are two types of concept drift: real and virtual. Real concept drift occurs because of sudden changes in a hidden context whereas virtual concept drift can occur when the target concept remains the same.

In real drift concept drift models need to be replaced because the former concept becomes invalid whereas in virtual concept drift models require further more learning as the failure of models may no longer be acceptable [15]. Fig. (1) shows the types of concept drift which represent instances and different classes.

Usually, virtual and real concept drift is seen to occur together, but there may be a case where virtual concept drift comes alone, for example, spam categorization [6]. In [26] sampling shift can be referred to virtual concept drift, and concept shift is referred to real concept drift shift. In practical terms, it is not crucial that real or virtual or both concept drift occurs, at the end of the result in all scenario the current model needs to be altered.

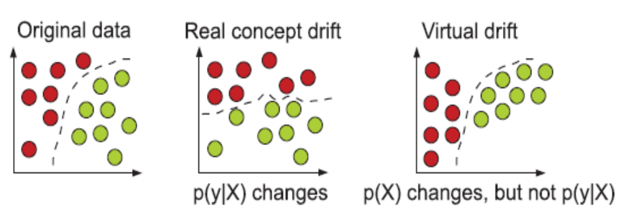

Fig. 1. Types of drifts: circles represent instances; different colors represent different classes [49]

Various changes can be observed in concept drift. Some regular types are gradual, sudden and recurring. The sudden altering is unforeseen when affecting the classification model. "For example, someone graduating from college might suddenly have completely different monetary concerns, whereas a slowly wearing piece of factory equipment might cause a gradual change in the quality of output parts.” [25]

Fig. (2) represents the pattern of changes in concept drift over time. The gradual changes evolve slowly through time. Gradual drift is divided into two categories: moderate and slow drifts, depending upon their rate of the changes by Stanley [25]. The recurring changes are a hidden context that reoccurs in two different manners: cyclically or in an unordered manner.

VI. METHODS FOR CONCEPT DRIFT

As stated in [28], to describe the knowledge and existence of concept drift, the algorithm designers must be able to recognize their main problem. The first problem is to detect the concept drift in the data stream, and once the concept drift is detected, the designer should be able to conclude the suitable process which will make a proper prediction for new upcoming data. To overcome the concept drift, there are some techniques which are divided into three main categories:

- Adaptive base learners
- Learners which modify
- Ensemble technique

A. STAGGER

STAGGER [8] was proposed by Schlimmer and Granger (1986b), it is the first concept drift handling system. STAGGER can create an order of data with both sudden and gradual concept drift and noise free example. By using STAGGER, synthetic data can be generated. STAGGER can be observed as a robust mechanism for dealing with noise and drift in learning. ”STAGGER concept is of Boolean function where three attributes encode the object such as **size** $\in$ (**small, medium, large**), **color** $\in$ (**red, blue, and green**), **shape** $\in$ (**circle, triangle, rectangle**). If any description which is covering green rectangle or red triangle, it will be denoted by (**shape = rectangle and color = green**) **or** (**shape = triangle and color =red**).” [8]

B. Decision Tree Based Method

Adaptive base learners can adapt the current training data which conflicts with the old data. By using adaptive base learners, concept drift can be quickly addressed. One base learner which is mostly researched is decision tree. VFDT (very fast decision tree) is the primary extension of the decision tree learning algorithm, which was proposed by Domingos and Hulten [31].

In VFDT, to grow the decision tree in streaming data, Hoeffding bounds [29], [30] are used in that the author considers that, in the data streaming scenario, applying Hoeffding bounds to the subset of data can choose the same split attribute as handling all of the data. Considering this view, decision trees can grow from a data stream, which is almost identical to a majority of all data.

There are many modifications of VFDT for data streams under concept drift. CVFDT (concept-adapting very fast decision tree) is one of them; it is a genuine expansion of VFDT which maintains the benefits of VFDT algorithm in classification accuracy [27]. In CVFDT, a sliding window of a sample is reserved in short term memory. When new samples arrive, after a fixed number of samples, old samples are removed, and new samples are added, and then Hoeffding bounds are again counted. If the better splitting attributes are found, the subtree identifies the concept drift occurred or not, and then new subtree is learned. The algorithm waits for more samples, and if new samples confirm that new subtree is learned better than the original one, then the original is replaced.

C. FLORA framework

According to the window size, the training set is prepared. The nave algorithm rarely keeps the fixed number of a new sample; because of this drawback, it is complicated to determine the suitable window size for any given problem. To overcome this problem, there are many approaches. One of the original windowing technique is proposed by Widmer and Kumbat [7,32] in FLORA3. FLORA3 is a modified version of FLORA [7] and FLORA2 by [33].

FLORA retains a dynamically adjusted time-window to trace the occurrence of concept drift. The FLORA framework is based on mutual conditions between the described item and the sample. In the training set, the rule-based concept description items are divided into the following three categories:

- Positive description set
- Negative description set
- Uncertain description set

FLORA updates the rule according to the entering and leaving time of the sample in the window, to move the sample from one set to another set or to resume them. FLORA has a limitation on the speed of the arriving data as it deals with only one sample at a time [9].

D. Meta-learning Methods

Regarding concept drift, in some domain changes of the concept are dependent on hidden context. The classification task of learning process provides explicit clues to the latest context. For this kind of domain, Widmer [38] proposed a dual-level learning model that can quickly adapt to the change in context by trying to detect via meta-learning contextual clues. The two operating systems are known as MetaL(B) and MetaL(IB). MetaL(B) is a combination of meta-learning, and Bayesian classifier and MetaL(IB) is based on the instance-based learning algorithm. These techniques can be trained to detect contextual clues a react whenever changes happen to the context [38]. When a new instance is introduced, it uses a set of attributes to establish new concepts. So that classifier can quickly detect the occurrence of concept drift. MetaL(IB) is another meta-learning strategy as MetaL(B). It uses instance-based learning algorithm as a base learner. In MetaL(IB), new incoming instances are classified by 1-NN (nearest neighbor) method. MetaL(IB) maintains a fixed size window.

E. ADWIN Methods

Bifet and Gevalda [11] have represented two approaches which are known as ADWIN and ADWIN2. These methods

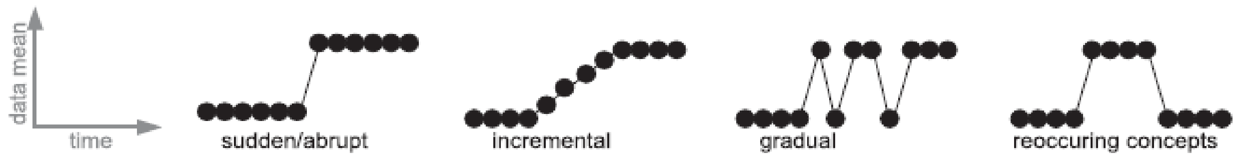

Fig. 2. Patterns of changes over time [49]

check all sub-windows of the latest window from the same distribution to determine the window size. ADWIN maintains a sliding window with its bits or real numbers. When there are no changes in data, the algorithm automatically grows and when the data changes it shrinks. Considering that ADWIN holds bits or real number, it has been used to work with the learning algorithm. i.e., to monitor the error rate of the current model. Because of this process, ADWIN takes more time and memory. ADWIN2 is proposed by using concepts of data stream algorithmics, which works on low memory and time requirements [11].

F. CD Algorithm

CD3 [39] is a detecting algorithm of concept drift. In this, every instance in the latest training set is assigned with a timestamp attribute. For every new data blocks, new timestamp attributes are assigned. It combines all data in the new data block and also combines all instances in the latest training set so that it can reconstruct the training set. Moreover, to reconstruct the classifier, it uses ID3 decision tree algorithm and post-pruning algorithm as base learner [9]. According to Hickey and Black [39], the concept drift occurs, when the time stamp attribute arrives in ID3 decision tree algorithm. CD4 and CD5 [40] are the extensions of CD3 respectively. Both methods cover the benefits of CD3 as they induce trees which keeps the record of the total history of changes. CD4 and CD5 produce larger tree compared to CD3 that is why it takes more time to learn.

G. Ensemble methods

The main task in data stream mining is to detect the occurrence of concept drift. In machine learning, there are lots of algorithms and tools available for the classification problem of concept drift. Ensemble classifier method is a leading stream for data mining classification method. For classification of a data stream, ensemble classifier method is more flexible and efficient; based on [46], ensemble algorithms are set of single classifier components whose decision are aggregated by a voting rule and the merged decision of abounding single classifiers is more definite than that given by a lone component. In the classification task, the ensemble classifier method selects some basic and easy classification algorithm like simple Bayes and decision tree.

Moreover, set the simple and primary classifier based on the samples. They try to get an ensemble classifier from the simple classifier, after getting the ensemble classifier they select the appropriate classifier or uses the combinations of simple classifiers to achieve the classification task [46]. Currently, the ensemble classifier follows this framework. To overcome the concept drift, there are various ensemble techniques based on different approaches.

1) SEA: An ensemble method which is proposed by Street and Kim is known as Streaming Ensemble Algorithm (SEA) [34]. This algorithm is a heuristic replacement strategy of a weak base classifier with majority voting which is based on two elements:

- Accuracy
- Diversity

SEA scores base classifier. It consists fixed number of base classifier ensemble frame, and when the base classifier in the frame gets the lower score, it replaces with another base classifier from current data blocks [34]. It has been observed by Street and Kim that, SEA performed pruned tree on static data sets and better on data sets with concept drift which results that it performs well with less than 25 components where base classifiers were unpruned and to combine the decisions, majority voting was used [48].

2) Accuracy Weighted Ensemble (AWE): Wang et al. [14] have proposed the weighted ensemble [WE] classifier model. It uses the ”eliminating the losers” method where current data blocks are used to test the base classifier. Whereas in AWE [35], the new classifier is trained with each new data chunk. These data chunks are used to access the ensemble members to select the best classifier. The AWE algorithm gives a better result with reoccurring concepts and different types of drift [35].

3) Accuracy Diversified Ensemble (ADE): Accuracy diversified ensemble algorithm selects the component and updates them as per the latest distribution. ADE [45] is different from AWE, such as in weight function, in bagging and upgrading the components with incoming data chunks. AWE was designed to create ensemble member from data chunks and only adjust the weight of components as per the latest distribution. In this concept, the size of the data chunk is an integral part of AWE which could be a drawback which can be solved by ADE [48].

4) Accuracy updated ensemble (AUE): AUE [45] handles the weighted pool of components and predicts the incoming data chunk by using majority weighted voting rules. After arriving of every data chunk, a new classifier is created, and the weak one gets replaced with a new classifier. The performance of each classifier is observed, and the weak performing component is updated. In short, their weight is adjusted as per the accuracy. For this process, it uses the Hoeffding tree as a component classifier. Compared to AWE it does not require cross-validation of the new classifier [48].

5) Hoeffding Option Trees and ASHT Bagging: These ensemble methods use Hoeffding tree for the best result. It allows the training data chunks to update its set of option node instead of single leaf. The author has combined the concepts of Hoeffding trees and option trees to introduce the Hoeffding option tree [36]. ASHT Bagging tree is Adaptive-Size Hoeffding Tree Bagging [37] which uses a forgetting mechanism and trees of different size to alter ensemble member.

VII. Issues and Outlooks

In machine learning, several methods are based on the classification task. While processing in a model, some algorithms are not able to provide a fair result because of some issues such as concept drift and noise, recurring of data and adjustment of training window size.

As reported in [6], While handling the concept drift, it is essential to know the actual concept drift and noise. There are some algorithms which are highly robust to noise and adjust the change slowly. Moreover, some algorithms which overreact to noise and by mistakenly consider it as concept drift. The learning classifier should be able to distinguish the concept drift from the noise, and it should be able to detect the presence of concept drift quickly. There must be such a quality to reuse the old concept and forecast the arriving concept. There are some learning classifiers which can reuse the data such as FLORA3 [7],[32], SPLICE [47].

VIII. Conclusion

This research describes main characteristics of classification technique of data stream with concept drift. As there are various types of changes occur in a data stream; some of the important types are described briefly. This paper is describing some parameters which are used to keep the classification model up-to-date for the data stream with concept drift. During the discussion, it focused on the importance of organic computing and describes how it is associated with concept drift.

It reviewed some approaches for handling concept drift like single base approach, sample-based approach, and explicit and implicit detection method. This study is focusing on the main methods to detect concept drift and predict the presence of concept drift. It is explaining the systems and methods to deal with concept drift such as STAGGER, meta-learning methods, CD Algorithms, FLORA framework, ADWIN methods, and some decision tree methods. Some ensemble classifier methods are also described briefly such as SEA, AWE, AUE, ADE and Hoeffding decision tree, and ASHT Bagging.

Finally, it sums up with issues which are faced by classification techniques and their outlooks and prospects to those problems to achieve sufficient result.

References

[1] Lee, Jeonghoon, and Yoon-Joon Lee. ”Concept Drift Detection for Evolving Stream Data.” IEICE TRANSACTIONS on Information and Systems 94.11 (2011): 2288-2292.

[2] Rangari, S. R., Dongre, S., and Malik, L. (2013). A new classifier for handling concept drifting data stream. International Journal of Science and Research, 2, 441-444.

[3] Kuncheva, Ludmila I. ”Classifier ensembles for detecting concept change in streaming data: Overview and perspectives.” In 2nd Workshop SUEMA, vol. 2008, pp. 5-10. 2008.

[4] Hand, David J. ”Classifier technology and the illusion of progress.” Statistical science (2006): 1-14.

[5] Gaber, M. M., and Yu, P. S. (2006). Classification of changes in evolving data streams using online clustering result deviation. In Proc. Of International Workshop on Knowledge Discovery in Data Streams.

[6] Tsymbal, Alexey. ”The problem of concept drift: definitions and related work.” Computer Science Department, Trinity College Dublin 106.2 (2004).

[7] Widmer, Gerhard, and Miroslav Kubat. ”Learning in the presence of concept drift and hidden contexts.” Machine learning 23.1 (1996): 69-101.

[8] Schlimmer, Jeffrey C., and Richard H. Granger. ”Incremental learning from noisy data.” Machine learning 1.3 (1986): 317-354.

[9] Zhenzheng, Ouyang, et al. ”Study on the classification of data streams with concept drift.” Fuzzy Systems and Knowledge Discovery (FSKD), 2011 Eighth International Conference on. Vol. 3. IEEE, 2011.

[10] Brzeziski, Dariusz. ”Mining data streams with concept drift.” Cs. Put. Poznan. Pl (2010): 89.

[11] Bifet, Albert, and Ricard Gavalda. ”Learning from time-changing data with adaptive windowing.” Proceedings of the 2007 SIAM international conference on data mining. Society for Industrial and Applied Mathematics, 2007.

[12] Klinkenberg, Ralf, and Ingrid Renz. ”Adaptive information filtering: Learning in the presence of concept drifts.” Learning for Text Categorization (1998): 33-40.

[13] Klinkenberg, Ralf. ”Learning drifting concepts: Example selection vs. example weighting.” Intelligent data analysis 8.3 (2004): 281-300.

[14] Wang, Haixun, et al. ”Mining concept-drifting data streams using ensemble classifiers.” Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. AcM, 2003.

[15] Beyene, Ayne A., et al. ”Improved concept drift handling in surgery prediction and other applications.” Knowledge and Information Systems 44.1 (2015): 177-196.

[16] Delany, Sarah Jane, Padraig Cunningham, and Alexey Tsymbal. ”A Comparison of Ensemble and Case-Base Maintenance Techniques for Handling Concept Drift in Spam Filtering.” FLAIRS Conference. 2006.

[17] Ganti, Venkatesh, Johannes Gehrke, and Raghu Ramakrishnan. ”Mining data streams under block evolution.” Acm Sigkdd Explorations Newsletter 3.2 (2002): 1-10.

[18] Markou, Markos, and Sameer Singh. ”Novelty detection: a reviewpart 1: statistical approaches.” Signal processing83.12 (2003): 2481-2497.

[19] Salganicoff, Marcos. ”Density-adaptive learning and forgetting.” Proceedings of the Tenth International Conference on Machine Learning. Vol. 3. 1993.

[20] Forman, George. ”Tackling concept drift by temporal inductive transfer.” Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2006.

[21] Littlestone, Nick. ”Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm.” Machine learning2.4 (1988): 285-318.

[22] Mesterharm, Chris. ”Tracking linear-threshold concepts with winnow.” Journal of Machine Learning Research 4.Oct (2003): 819-838.

[23] Fan, Wei, Salvatore J. Stolfo, and Junxin Zhang. ”The application of AdaBoost for distributed, scalable and on-line learning.” Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 1999.

[24] Fan, Wei. ”Streamminer: A classifier ensemble-based engine to mine concept-drifting data streams.” Proceedings of the Thirtieth international conference on Very large data bases-Volume 30. VLDB Endowment, 2004.

[25] Stanley, Kenneth O. ”Learning concept drift with a committee of decision trees.” Informe tcnico: UT-AI-TR-03-302, Department of Computer Sciences, University of Texas at Austin, USA (2003).

[26] Salganicoff, Marcos. ”Tolerating concept and sampling shift in lazy learning using prediction error context switching.” Lazy learning. Springer, Dordrecht, 1997. 133-155.

[27] Hulten, Geoff, Laurie Spencer, and Pedro Domingos. ”Mining time-changing data streams.” Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2001.

[28] Hoens, T. Ryan, Robi Polikar, and Nitesh V. Chawla. ”Learning from streaming data with concept drift and imbalance: an overview.” Progress in Artificial Intelligence 1.1 (2012): 89-101.

[29] Maron, Oded, and Andrew W. Moore. ”Hoeffding races: Accelerating model selection search for classification and function approximation.” Advances in neural information processing systems. 1994.

[30] Hoeffding, Wassily. ”Probability inequalities for sums of bounded random variables.” Journal of the American statistical association 58.301 (1963): 13-30.

[31] Domingos, P., and Hulten, G. (2000, August). Mining high-speed data streams. In Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 71-80). ACM.

[32] Widmer, Gerhard, and Miroslav Kubat. ”Effective learning in dynamic environments by explicit context tracking.” European Conference on Machine Learning. Springer, Berlin, Heidelberg, 1993.

[33] Widmer, Gerhard, and Miroslav Kubat. ”Learning flexible concepts from streams of examples: FLORA2.” (1992).

[34] Street, W. N., and Kim, Y. (2001, August). A streaming ensemble algorithm (SEA) for large-scale classification. In Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 377-382). ACM.

[35] Elwell, R., and Polikar, R. (2011). Incremental learning of concept drift in nonstationary environments. IEEE Transactions on Neural Networks, 22(10), 1517-1531.

[36] Pfahringer, Bernhard, Geoffrey Holmes, and Richard Kirkby. ”New options for hoeffding trees.” Australasian Joint Conference on Artificial Intelligence. Springer, Berlin, Heidelberg, 2007.

[37] Bifet, Albert, et al. ”New ensemble methods for evolving data streams.” Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2009.

[38] Widmer, Gerhard. ”Tracking context changes through meta-learning.” Machine Learning 27.3 (1997): 259-286.

[39] Black, Michaela, and Ray J. Hickey. ”Maintaining the performance of a learned classifier under concept drift.” Intelligent Data Analysis 3.6 (1999): 453-474.

[40] Hickey, Ray J., and Michaela M. Black. ”Refined time stamps for concept drift detection during mining for classification rules.” Temporal, Spatial, and Spatio-Temporal Data Mining. Springer, Berlin, Heidelberg, 2001. 20-30.

[41] Tomforde, Sven, et al. ”Observation and control of organic systems.” Organic ComputingA Paradigm Shift for Complex Systems. Springer, Basel, 2011. 325-338.

[42] Tomforde, Sven, Bernhard Sick, and Christian Mller-Schloer. ”Organic Computing in the Spotlight.” arXiv preprint arXiv:1701.08125 (2017).

[43] Mller-Schloer, Christian. ”Organic computing: on the feasibility of controlled emergence.” Proceedings of the 2nd IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis. ACM, 2004.

[44] Seebach, Hella, Frank Ortmeier, and Wolfgang Reif. ”Design and construction of organic computing systems.” Evolutionary Computation, 2007. CEC 2007. IEEE Congress on. IEEE, 2007.

[45] Brzezinski, D., and Stefanowski, J. (2014). Reacting to different types of concept drift: The accuracy updated ensemble algorithm. IEEE Transactions on Neural Networks and Learning Systems, 25(1), 81-94.

[46] Kuncheva, L. I. (2004, June). Classifier ensembles for changing environments. In International Workshop on Multiple Classifier Systems (pp. 1-15). Springer, Berlin, Heidelberg.

[47] Harries, M. B., Sammut, C., and Horn, K. (1998). Extracting hidden context. Machine learning, 32(2), 101-126.

[48] P. B. Dongre, L. G. Malik(2014), ” A Review on Real Time Data Stream Classification and Adapting To Various Concept Drift Scenarios”, IEEE International Advance Computing Conference (IACC), pp. 533-537.

[49] Gama, Joäo, et al. ”A survey on concept drift adaptation.” ACM computing surveys (CSUR) 46.4 (2014): 44.