A Review on Dynamic Concept Drift

D.Kishore Babu [1,] Dr. Y.Ramadevi [2], Dr.K.V.Ramana[3]

1. Dept of CSE, BVCITS, Amalapuram, Andhra Pradesh
2. Prof in the Dept of CSE, CBIT, Gandi Pet, Hyderabad, Telanga State
3. Prof in the Dept of CSE, JNTU Kakinada, Andhra Pradesh

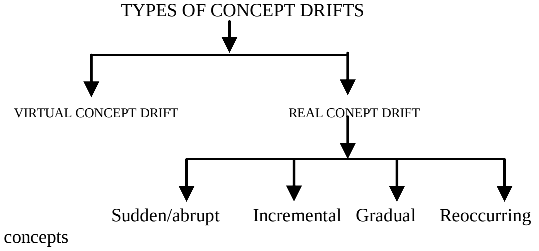
ABSTRACT

In the real world concepts are frequently not stable but change with time. Typical examples of this are weather prediction rules and customers' preferences. The underlying data distribution may change as well. Over and over again these changes make the model built on old data inconsistent with the new data, and regular updating of the model is necessary. This problem, known as concept drift, complicates the task of learning a model from data and requires special approaches, different from commonly used techniques, which treat arriving instances as equally important contributors to the final concept. Dynamic concept drift is one of the techniques which will handle the concept drift effectively by using ensembles. Dynamic concept drift with Naïve base classifier is finest one.

Keywords: *concept drift, ensembles, dynamic integration, dynamic weighted majority.*

INTRODUCTION

In the real world concepts and data distributions are often not stable but change with time. This problem, known as concept drift *[1]*. Concept drift occurs when the concept about which data is being composed shifts from time to time after a minimum stability period. This problem of concept drift needs to be considered to mine data with acceptable accuracy level. Some examples of concept drift include spam detection, financial fraud detection, climate change prediction, customer preferences for online shopping. A difficult problem with learning in many real-world domains is that the concept of interest may depend on some hidden context, not given explicitly in the form of predictive features. *[2]* .Concept drifts are of two types; those are virtual and real,



Depending on the relation between the input data and target variable, concept change take different forms. Concept drift between time point t0 and time point t1 can be defined as-

$$\exists X: pt0(X, y) \neq pt1(X, y) \qquad (3)$$

Where pt0 denotes the joint distribution at time t0 between the set of input variables X and the target variable y. Kelly et al. presented the three ways in which concept drift may occur [4]:

- Prior probabilities of classes, p(y) may change over time
- Class-conditional probability distributions, p(X,y) might change
- Posterior probabilities p (y|X) might change.

Concept drift may be classified in terms of the [5] speed of change and the reason of change as shown in figure 1. When 'a set of examples has legitimate class labels at one time and has different legitimate labels at another time', it is real drift, i.e. reason of change[6], refers to changes in p(y|X).

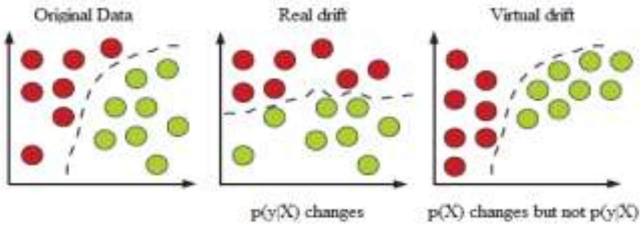


Fig 1**:** Types of drift: circles represent instances; different colors represent different classes [7]

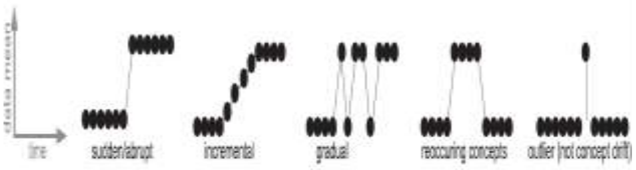


Fig 2**:** Patterns of concept changes [7]

When 'the target concepts remain the same but the data distribution changes’ [8], it is virtual drift, i.e. speed of change, refers to changes in p(X). A drift can be sudden or abrupt, when concept switching is from one to another (refer figure 2) [7]. The concept change can be incremental, consisting of many intermediate concepts in between. Drift may be gradual; change is not abrupt, but goes back to previous pattern for some time. Concept drift handling algorithms should not mix the true drift with an outlier (blip) or noise, which refers to an anomaly. A recurring drifts is when new concepts that were not seen before, or previously seen concepts may reoccur after some time.

Changes in hidden context may not only be a cause of a change of target concept, but may also cause a change of the underlying data distribution. [9].Changes in the unseen context can bring more or less drastic changes in the target concept [10]. A difficult problem in handling concept drift is distinguishing between true concept drift and noise. Some algorithms may overreact to noise, erroneously interpreting it as concept drift, while others may be highly robust to noise, adjusting to the changes too slowly. An ideal learner should combine robustness to noise and sensitivity to concept drift [10].In several domains, hidden contexts may be expected to happen again. Frequent contexts may be due to repeated phenomena, such as seasons of the year or may be associated with irregular phenomena, such as inflation rates or market mood [11].

ENSEMBLES:

Ensemble learning can be used for improving prediction accuracy. This is what made ensemble learning the most common technique that has been used to deal with the concept drift issue. In ensemble learning, the final decision must be taken either by combined classifications outputs of several models or by selecting the best one [12].

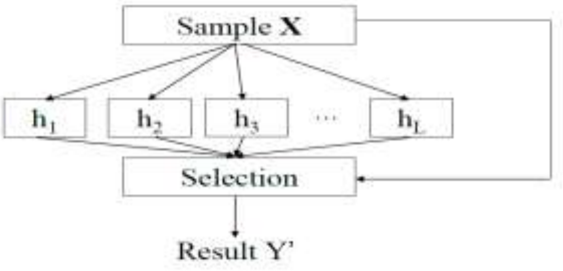


Fig 3: Ensemble learning system architecture [13]

In Figure 1, we present the most commonly used architecture of ensemble classifiers. In this example, represents the type of ensemble classifier that we use, such as bagging; while stands for the individual classifier that we use, such as decision trees. After that, each ensemble classifier has a mechanism to select the best classifier output or combine more than one classifier output [14].Ensemble is a group of items viewed as a whole rather than individually. An ensemble consists of a set of individually trained classifiers whose predictions are combined when classifying novel instances. Ensemble modeling is the process of running two or more related but different analytical models and then synthesizing the results into a single score or spread in order to improve the accuracy of predictive analytics and data mining applications. Ensemble learning is a powerful method to utilize the diversity of machine learning methods. An ensemble of learners is potentially able to outperform every one of its members. While in general this is well-known, the full power of ensembles, especially in applications.

Ensemble learning is the most popular and effective approach to handle concept drift the approach to handle concept drifts includes single classifier and ensemble classifier approaches. The single classifiers are traditional learners that were modeled for stationary data mining and have the qualities of an online learner and a forgetting mechanism. Basically, ensemble classifiers are sets of single classifiers whose individual decisions are aggregated by a voting rule. The ensemble classifiers provide better classification accuracy as compared to the single classifiers due combined decision. They have a natural way of adapting to concept changes due to their modularity. [15].

Ensemble learning is the most popular and effective approaches to handle concept drift, in which a set of concept descriptions built over different time intervals is maintained, predictions of which are combined using a form of voting, or the most relevant description is selected.[16].There are different Methods offered in Ensembles:

Streaming Ensemble Algorithm (SEA): The SEA [17], proposed by Street and Kim, changes its structure based on concept change. It is a heuristic replacement strategy of the weakest base classifier based on accuracy and diversity. The combined decision was based on simple majority voting and base classifiers unpruned. This algorithm works best for at most 25 components of the ensemble. simple majority voting was used to combine member decisions The main weakness of Streaming Ensemble Algorithm is it will not hold up for large data streams. It adapts to gradual changes, but it has trouble adapting to abrupt concept drifts.

Accuracy Weighted Ensemble (AWE): In SEA, it is crucial to properly define the data chunk size as it determines the ensembles flexibility. The algorithm AWE, proposed by Wang et al., trains a new classifier C' on each incoming data chunk and use that chunk to evaluate all the existing ensemble members to select the best component classifiers. AWE is best suited for large data streams and works well for recurring and other drifts. The AWE algorithm works well on data streams with reoccurring concepts as well as different types of drift. As with SEA it is crucial to properly define the data chunk size as it determines the ensembles flexibility. It is also worth noticing, that AWE will improve its performance gradually over time and is best suited for large data streams.

Adaptive Classifier Ensemble (ACE): To overcome AWE‟ s slow drift reactions, Nishida proposed a hybrid approach in which a data chunk ensemble is aided by a drift detector, called Adaptive Classifier Ensemble (ACE), aims at reacting to sudden drifts by tracking the classifier‟ s error rate with each incoming example, while slowly reconstructing a classifier ensemble with large chunks of examples.

Hoeffding option trees (HOT) and ASHT Bagging: Hoeffding Option Trees (HOT) provide a compact structure that works like a set of weighted classifiers, and are built in an incremental fashion. This algorithm [18] allows each training example to update a set of option nodes rather than just a single leaf. Adaptive-Size Hoeffding Tree Bagging (ASHT Bagging) diversifies ensemble members by using trees of different sizes and uses a forgetting mechanism. Compared to HOT, ASHT Bagging proves to be more accurate on most data sets. But both are time and memory expensive than option trees or single classifiers.

Accuracy Diversified Ensemble (ADE): The algorithm called Accuracy Diversified Ensemble (ADE) [19], not only selects but also updates components according to the current distribution. ADE differs from AWE in weight definition, the use of online base classifiers, bagging, and updating components with incoming examples. Compared to ASHT and HOT, we do not limit base classifier size, do not use any windows, and update members only if they are accurate enough according to the current distribution.

Accuracy Updated Ensemble (AUE): Compared to AWE, AUE1conditionally updates component classifiers. It maintains a weighted pool of component classifiers and predicts classes for incoming examples based on weighted voting rule. It substitutes the weakest performing ensemble member and new classifier is created with each data chunk of examples, also their weights are adjusted. It uses Hoeffding trees as component classifiers. Compared to AUE1, AUE2 introduces a new weighting function[19], does not require cross-validation of the candidate classifier, does not keep a classifier buffer, prunes its base learners, and always updates its components. It does not limit base classifier size and use any windows. The OAUE [20], tries to combine block-based ensembles and online processing.

We can build ensemble well by using sliding window approach, to build ensembles through sliding window, we divide the data into blocks corresponding to a certain time interval. We use the sliding window approach, and thus, when the window shift is less than the size of the window, the data blocks are not mutually exclusive. We use the last (current) data block as the test set, and the current ensemble includes only those base classifiers that are built on preceding data blocks that include different instances only with regard to the test set in order to avoid overly optimistic error estimate for the ensemble [21], the main drawback of window-based approaches with local concept drift is that instances with different relevance with regard to the current concept are often included in the window. [22] There is a problem with current ensemble approaches in that they are not able to deal with local concept drift, which is a common case with real-world data. [23].if ensembles are more it takes much more time to process and if ensembles are less then we can perform in timely, sometimes there is a chance of occurring similar ensembles.

DYNAMIC INTEGRATION

Dynamic integration can be a more appropriate integration technique for handling concept drift, and that it may be especially useful in the presence of local concept drift [24].In dynamic integration each new instance to be classified is taken into account. Usually, better results can be achieved if integration is dynamic. Dynamic integration approach for ensembles used in tracking concept drift, which integrates the base classifiers at instance level, assigning to them weights proportional to their local accuracy on each instance considered [25], Dynamic integration techniques improve ensemble accuracy The challenging problem of integration is to decide which of the classifiers to select or how to combine the results produced by the base classifiers. A number of selection and combination approaches have been proposed**.** [26]

One of the most popular and simplest techniques used to combine the results of base classifiers, is simple voting (also called majority voting) [27]. In voting, the output of each base classifier is considered as a vote for that particular class value. The class value that receives the biggest number of votes is selected as the final classification. Weighted Voting (WV), where each vote has a weight proportional to the estimated generalization performance of the corresponding classifier, usually works better than the simple voting. [28]

A number of selection techniques have also been proposed to solve the integration problem. One of the most popular and simplest selection techniques is Cross-Validation Majority (CVM, also called Single Best)[29]. In CVM, cross-validation accuracy for each base classifier is estimated, and then the classifier with the highest accuracy is selected. The approaches described above are stationary. They choose one model for the entire instance space or join the models uniformly. In dynamic integration each new instance to be classified is taken into account. Usually, better results can be achieved if integration is dynamic. [30], [31], [32], [33], [34], [35].Three dynamic techniques based on the same local error estimates: Dynamic Selection (DS), Dynamic Voting (DV), and Dynamic Voting with Selection (DVS).In DS a classifier with the least predicted local classification error is selected. In DV, each base classifier receives a weight that is proportional to its estimated local accuracy and the final classification is produced as in WV. In DVS, the base classifiers with the highest local classification errors are discarded. [36]

WEIGHTED MAJORITY

Weighted Majority algorithm is one of the method which will hold the concept drift, Weighted Majority algorithm is a general method based on the Weighted Majority algorithm for using any online learner for concept drift. It is a method for weighting and combining the decisions of experts or base learners, each of which is a learning method. The algorithm begins by creating a set of experts and assigning a weight to each.

Bagging is one of the simplest ensemble method [38]; Bagging creates multiple predictors and uses them to get an aggregated predictor as shown in Figure 2. For an estimation problem, Bagging does aggregation averages over the multiple predictors while for classification problem, Bagging does majority voting to predict a class. Bagging makes bootstrap replicates of the learning set and uses them as new learning sets to form the multiple predictors. In other words, based on a uniform probability distribution, Bagging samples the training sets with replacement. Every sample has the same size as the original data, some instances may appear more than once in the same training set and then bagging builds a classifier on each sample [39] With bagging, when one new example arrives that is misclassified, it is too inefficient to resample the available data and learn new classifiers.

Boosting, which is popular ensemble learning, is a general method for increasing the accuracy of any learning method. Freund and Schapire [40] introduced the AdaBoost algorithm in 1995. Boosting is a sequential production of classifiers where each classifier focuses on the previous one’s errors. Each classifier focuses on the examples that are incorrectly predicted in previous classifiers by weighting them more [40].

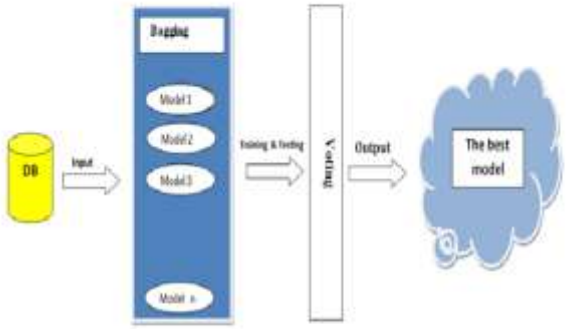


Fig: 4 Bagging [41]

DYNAMIC WEIGHTED MAJORITY

DWM is based on the Weighted Majority Algorithm(WMA) Dynamic Weighted Majority (DWM) is a New Ensemble Method for Concept Drift, DWM maintains an ensemble of base learners, predicts using a weighted-majority vote of these “experts”, and dynamically creates and deletes experts in response to changes in performance. Dynamic Weighted Majority (DWM) maintains as its concept description an ensemble of learning algorithms, each referred to as an expert and each with an associated weight. Given an instance, the performance element polls the experts, each returning a prediction for the instance. Using these predictions and expert weights, DWM returns as the global prediction the class label with the highest accumulated weight, In DWM each classifier’s weight is determined by its error, age, and performance on current and all previous environments

DWM is based on the Weighted Majority Algorithm (WMA) [42] which takes the idea of working with a group of experts, to which an initial weight is automatically assigned. Then, when a new example arrives, the base algorithm receives a prediction from each expert and makes a final decision by combining the predictions and the weights of each expert; finally, if an expert makes an incorrect prediction, then its weight is reduced by a multiplicative constant between 0 and 1. In order to adapt to working with data streams and to handle concept drifts, DWM includes mechanisms to add, update, and delete base classifiers. A test is performed and a new classifier is added with a weight value equal to 1 if the system output is incorrect; moreover, the system deletes each base classifier, whose weight falls below a threshold of θ. One of the potential problems of this algorithm is that it penalizes base classifiers when they fail but it does not reward them when they are right; this makes the base classifiers' weights fall quickly and they only remain a short while within the ensemble; this, coupled with the fact that DWM steadily updates the base classifiers, the difficulty of DWM is does not make it suitable for the treatment of recurring concepts [43].

We have evaluated two base learner algorithms for DWM: DWM-NB (*naive Bayes)* and DWM-ITI (Incremental Tree Inducer). An online version of naïve bases stores counts for the number of examples processed, the number of examples of each class, and the number of attribute values given each class. Learning entails incrementing these counts as new examples arrive. To classify an observation, the performance element uses these counts to compute estimates of the prior and class-conditional probabilities, assume conditional independence of the attributes, and uses Bayes rule to determine the most probable class.

The performance element uses these counts to compute estimates of the prior probability of each class, $P(Ci)$, and the conditional probability of each attribute value given the class, $P(v\ j|Ci)$. It then operates under the assumption that attributes are conditionally independent and uses Bayes' rule to predict the most probable class:

$$C = \underset{Ci}{\operatorname{argmax}}\ P\,(Ci)\ \prod_{j} P\,(v\ \mathrm{j}|Ci),$$

For numeric attributes, it stores the sum of an attribute's values and the sum of the squared values. Given a value, *v j*,

$$e\square\ (v\,j\text{-}\mu i\,j)^{2}/2\sigma^{2}ij,$$

$$P\,(v\,j|Ci) = 1\square\ \sigma i\,j\sqrt{2}$$

Where µi *j* is the average of the *j*th attribute's values for the *i*th class and σ*i j* is their standard deviation. The performance element computes these values from the stored sums [34]

DWM-NB having more experts and it takes much more time. ITI is an incremental algorithm for inducing decision tree. A tree is a rooted tree with internal nodes that correspond to attributes and external nodes that correspond to class labels. From an internal, attribute node, there are edges for each value the attribute takes.ITI produces decision trees with only binary splits, and at each node, it maintains a set of counts of class label and attribute values. DWM-ITI having less experts and it takes less time to process Concepts. Results on these problems, when compared to other methods, suggest that DWM maintained a comparable number of experts, but achieved higher predictive accuracies and converged to those accuracies more quickly. Indeed, to the best of our knowledge, these are the best overall results reported for these problems.

CONCLUSIONS AND FUTURE WORK

Dynamic weighted majority with naïve base is one of the system which will handle concept drift, In future exertion, we plan to explore additional sophisticated heuristics for better results, and perhaps DWM should take into account the expert's age or its history of predictions. We would also like to investigate another decision tree learner as a base algorithm; we can reduce usage of memory as change can be take place and we have to scrutinize tracking recurring concept drifts which are available in the real world Streaming data base algorithm. And we can reduce usage of memory as change can be take place.

REFERENCES

[1].Alexey Tsymbal, Mykola Pechenizkiy, Pádraig Cunningham, Seppo Puuronen,"Handling Local Concept Drift with Dynamic Integration of Classifiers: Domain of Antibiotic Resistance in Nosocomial Infections", Proceedings of the 19th IEEE Symposium on Computer-Based Medical Systems (CBMS'06) 0-7695-2517-1/06, 2006 IEEE.
[2].Alexey Tsymbal Department of Computer Science Trinity College Dublin, Ireland, "The problem of concept drift: definitions and related work ", April 29, 2004.TCD-CS-2004-15.

[3].P. M. Goncalves, Silas G.T. de Carvalho Santos, Roberto S.M. Barros, Davi C.L. Vieira, (2014) "Review: A comparative study on concept drift detectors", A International Journal: Expert Systems with Applications, 8144–8156.

[4].M. G. Kelly, D. J. Hand, and N. M. Adams(1999), "The Impact of Changing Populations on Classifier performance", In Proc. of the 5th ACM SIGKDD Int. Conf. on Knowl. Disc. and Dat. Mining (KDD). ACM, 367–371.

[5]. J. Gama, I. Zliobaite, A. Bifet, M. Pechenizkiy, A. Bouchachia(2014), "A Survey on Concept Drift Adaptation ", ACM Computing Surveys, Vol. 46, No. 4, Article 44.

[6]. J. Kolter and M. A. Maloof (2007), "Dynamic Weighted Majority: An Ensemble Method for Drifting Concepts", Journal of Machine Learning Research 8, 2755-2790.

[7]. J. Gama, I. Zliobaite, A. Bifet, M. Pechenizkiy, A. Bouchachia(2014), "A Survey on Concept Drift Adaptation ", ACM Computing Surveys, Vol. 46, No. 4, Article 44.

[8]. S. Delany, P. Cunningham, A. Tsymbal, and L. Coyle. (2005),"A Case-based Technique for Tracking Concept Drift in Spam filtering", Knowledge-Based Sys. 18, 4–5, 187–195

[9].Alexey Tsymbal, Mykola Pechenizkiy, Pádraig Cunningham, Seppo Puuronen,"Handling Local Concept Drift with Dynamic Integration of Classifiers: Domain of Antibiotic Resistance in Nosocomial Infections", Proceedings of the 19th IEEE Symposium on Computer-Based Medical Systems (CBMS'06) 0-7695-2517-1/06, 2006 IEEE.

[10].Widmer G., Kubat M., Learning in the presence of concept drift and hidden contexts, Machine Learning, 23 (1), 1996, 69-101.

[11]. Harries M., Sammut C., Horn K., Extracting hidden context, Machine Learning, 32(2),1998, 101-126).

[12].Indre Zliobaite, "Learning under Concept Drift: an Overview". Lithuania: Faculty of Mathematics and Informatics, Vilnius University, Lithuania.

[13].D. Yeung and P. Chan. (2006). MCS Tutorial [Online]. www.ece.stevens-tech.edu/~hhe/cpe695f09/lecturenotes.

[14].Mohammed Alshammeri, "Dynamic Committees for Handling Concept Drift in Databases (DCCD)", School of Electrical Engineering and Computer Science University of Ottawa, Canada 2012.

[15]. Yamini Kadwe, Vaishali Suryawanshi Department of IT, M.I.T. College Of Engineering, Pune, India. "A Review on Concept Drift". IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p-ISSN: 2278-8727, Volume 17, Issue 1, Ver. II (Jan – Feb. 2015), PP 20-26.

[16].Alexey Tsymbal, Mykola Pechenizkiy, Pádraig Cunningham, Seppo Puuronen,"Handling Local Concept Drift with Dynamic Integration of Classifiers: Domain of Antibiotic Resistance in Nosocomial Infections", Proceedings of the 19th IEEE Symposium on Computer-Based Medical Systems (CBMS'06) 0-7695-2517-1/06, 2006 IEEE.

[17]. W. N. Street and Y. Kim (2001), "A streaming ensemble algorithm (SEA) for large-scale classification," in Proc. 7th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, pp. 377–382.

[18]. A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, and R. Gavaldà(2009),"New ensemble methods for evolving data streams," in Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, pp. 139-148.

[19]. D Brzezinski, J Stefanowski (2014),"Reacting to Different Types of Concept Drift: The Accuracy Updated Ensemble Algorithm" IEEE Transactions on Neural Networks and Learning Systems," Vol. 25, pp. 81-94.

[20]. D Brzezinski, J Stefanowski(2014), "Combining block-based and online methods in learning ensembles from concept drifting data streams", An International Journal: Information Sciences 265, 50–67.

[21].Alexey Tsymbal, Mykola Pechenizkiy, Pádraig Cunningham, Seppo Puuronen,"Dynamic Integration of Classifiers for Handling Concept Drift", Journal, Information Fusion Volume 9 Issue 1, January, 2008, Pages 56-68, Elsevier Science Publishers.

[22].Alexey Tsymbal, Mykola Pechenizkiy, Pádraig Cunningham, Seppo Puuronen,"Dynamic Integration of Classifiers for Handling Concept Drift."Journal, Information Fusion Volume 9 Issue 1, January, 2008, Pages 56-68, Elsevier Science Publishers.

[23]. Alexey Tsymbal, Mykola Pechenizkiy, Pádraig Cunningham, Seppo Puuronen,"Dynamic Integration of Classifiers for Handling Concept Drift", Journal, Information Fusion Volume 9 Issue 1, January, 2008, Pages 56-68, Elsevier Science Publishers.

[24]. Alexey Tsymbal, Mykola Pechenizkiy, Pádraig Cunningham, Seppo Puuronen. "Dynamic Integration of Classifiers for Handling Concept Drift", Journal, Information Fusion Volume 9 Issue 1, January, 2008, Pages 56-68, Elsevier Science Publishers.

[25]. Alexey Tsymbal1, Mykola Pechenizkiy, Pádraig Cunningham1, Seppo Puuronen. "Dynamic Integration of Classifiers for Tracking Concept Drift in Antibiotic Resistance Data", Journal, Information Fusion Volume 9 Issue 1, January, 2008, Pages 56-68, Elsevier Science Publishers.

[26]. Alexey Tsymbal, Mykola Pechenizkiy, Pádraig Cunningham, Seppo Puuronen, "Dynamic Integration of Classifiers for Handling Concept Drift", Journal, Information Fusion Volume 9 Issue 1, January, 2008, Pages 56-68, Elsevier Science Publishers.

[27].Bauer E., Kohavi R. "An empirical comparison of voting classification algorithms: bagging, boosting, and variants", Machine Learning 36, 105–139 (1999) 1999 Kluwer Academic Publishers. Manufactured in The Netherlands.

[28].Jeremy Z. Kolter and Marcus A.Maloof Department of Computer Science Georgetown University "Dynamic Weighted Majority: A New Ensemble Method for Tracking Concept Drift", Proceedings of the Third International IEEE Conference on Data Mining, 123–130.Los Alamitos, CA: IEEE Press.

[29]. J. Zico Kolter, Marcus A. Maloof**,"**Dynamic weighted majority: an ensemble method for drifting concepts", Journal of Machine Learning Research 8 (2007) 2755-2790, Third IEEE International Conference on Data Mining, 2003 IEEE.

[30].Leo. Breiman, Statics Department University of California Berkeley, CA94720, "Bagging Predictors," Machine Learning, vol. 24, pp. 123-140(1996) 1996 Kluwer Academic Publishers, Boston. Manufactured in the Netherlands.

[31].Yoav Freund and Robert E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," Journal of Computer and System Sciences, vol. 55(1), pp 119–139, August 1997

[32].N. Littlestone and M. K. Warmuth, Department of computer science ,University of California, santa Cruz, California 95064,"The weighted majority algorithm," Information and Computation, vol. 108, pp.212–261, 1994.

[33].Agustín Ortiz Díaz, José del Campo-Ávila, Gonzalo Ramos-Jiménez, Isvani Frías Blanco,Yailé Caballero Mota, Antonio Mustelier Hechavarría, and Rafael Morales-Bueno, "Fast Adapting Ensemble: A New Algorithm for Mining Data Streams with Concept Drift", Hindawi, Publishing Corporation The Scientific World Journal, Volume 2015, Article ID 235810, 14 pages.

[34]. J. Zico Kolter, Marcus A. Maloof,"Dynamic weighted majority: an ensemble method for drifting concepts", Journal of Machine Learning Research 8 (2007) 2755-2790, Third IEEE International Conference on Data Mining, 2003 IEEE.

[35].K.Woods,W.P. Kegelmeyer, K. Bowyer, "Combination of multiple classifiers using local accuracy estimates", IEEE Transaction on PAMI, 19 (4), 1997, 405-410

[36].Alexey Tsymbal, Mykola Pechenizkiy, Pádraig Cunningham, Seppo Puuronen "Dynamic Integration of Classifiers for Tracking Concept Drift in Antibiotic Resistance Data", Information Fusion Volume 9 Issue 1, January, 2008, Pages 56-68, Elsevier Science Publishers

[37].Bauer E., Kohavi R. "An empirical comparison of voting classification algorithms: bagging, boosting, and variants", Machine Learning 36, 105–139 (1999) 1999 Kluwer Academic Publishers. Manufactured in The Netherlands.

[38]. Cullen Schaffer "Selecting a classification method by cross-validation", Machine Learning, 13, 135-143 (1993) 1993, Kluwer Academic Publishers, Boston. Manufactured in the Netherlands.

[39].G.Giacinto, F. Roli, "Dynamic classifier selection based on multiple classifier behavior," Pattern Recognition, 34 (9), (2001), 1879-1881. Dept. of Electrical and Electronic Eng, Univ. of Cagliari, Piazza d'Armi, 09123 Cagliari, ITALY.

[40].G.Giacinto, F. Roli, "Methods for dynamic classifier selection," In: ICIAP '99, 10th International Conference on Image Analysis and Processing, Venice, Italy, IEEE CS Press, September,1999, pp.659-664.

[41]. Mohammed Alshammeri, "Dynamic Committees for Handling Concept Drift in Databases (DCCD)", School of Electrical Engineering and Computer Science University of Ottawa, Canada 2012.

[42].Robnik-Sikonja M. "Improving random forests". In: ECML 2004 J.F. 15th European Conf. on Machine Learning Pisa Italy, September 2004, 2004, pp.359-370.

[43]. Tsymbal A., M. Pechenizkiy, P. Cunningham, "Sequential genetic search for ensemble feature selection." In: Proc. 19th Int. Joint Conf. on Artificial Intelligence IJCAI'2005, Morgan Kaufmann, August 2005, pp.877-882.

[44]. Tsymbal A., Puuronen S. "Bagging and boosting with dynamic integration of classifiers." In: PKDD 2000,Principles of Data Mining and Knowledge Discovery, Lyon, France, September 2000, pp.116-125.