CONCEPT DRIFT IN DATA STREAM CLASSIFICATION USING ENSEMBLE METHODS: TYPES, METHODS AND CHALLENGES

MS. THANGAM M[1] AND DR. A. BHUVANESWARI[2]

Department of Computer Science,
Cauvery College for Women (Autonomous), [Affiliated to Bharathidasan University], Tiruchirappalli, India
Email: thangamm.it@cauverycollege.ac.in
Department of Computer Science,
Cauvery College for Women (Autonomous), [Affiliated to Bharathidasan University], Tiruchirappalli, India
Email: bhuvaneswari.it@cauverycollege.ac.in

Abstract - Ensemble Methods grows along with Machine Learning and Computational Intelligence domain proves to be effective and versatile. It helps in reducing variance and improves accuracy. Few Machine Learning challenges such as data stream classification, class Imbalance in datasets and occurrence of concept drift in non-stationary environments is addressed effectively by Ensemble methods. Data stream refers to rapidly generated heterogeneous data in a continuous way. One of the key challenges considered in learning from data streams is the detection of concept drift, i.e., changes in data distribution underlying data streams, observed over time. Such changes in incoming data deteriorate the accuracy of the classifier since classifier has been learned over past data instances that are stable. Thus detection of concept drift is an important task. The real life examples of drift are spam detection, credit card fraud detection, and weather predictions. This paper presents a survey on highlighting recent research ideas in concept drift using ensemble methods and also provides a comprehensive introduction to ensemble methods, data stream classification models, types of concept drift and drift detection methods.

Keywords: Ensemble, Classifier, Data Streams, Concept drift, Sudden Drift, Gradual Drift, Drift Detection

(Received July 1st, 2020 / Accepted December 1st, 2020)

Introduction

In the domain of Machine Learning, research is focused on batch learning using small datasets where the algorithm uses the entire training data and predicts a model for making decisions after undergoing multiple processing [10]. With the recent technological advancement, large amounts of data are generated with increase in the number of applications that includes online shopping, intrusion detection, traffic control etc. The focus of research is intended to analyze huge volumes of data to make appropriate business decisions based on the knowledge hidden in stored data and thereby the learning algorithms need to act in dynamic environments in which data are collected in the form of data streams [2].

Data Stream refers to continuous, potentially unbounded, ordered and infinite sequence of heterogeneous data items with a high generation rate that arrives over time [1]. Data items may be simple or complex. It is a sequence of labeled examples and generates instances that appear as a sequence in time intervals [7]. Data streams distinguishes from conventional data with the following unique features like immense and infinite volume, sequential arrival of data items over time, high generation rate, rapid arrival rate, susceptible to change and data labeling may be costly in data streams [1][3].

The two models of data streams are: Stationary data streams where data items are generated from fixed, probability distribution [10] and Non-stationary or evolving data streams are time series data that evolves over time [10][21]. In streaming data classification, the dataset size is unknown due to the evolving nature of streaming data and poses several challenges that includes temporal dependencies, huge instances, limited labeled instances, novel classes, feature drifts, resource constraints(processing time, memory), single scan of the data (one look, no random access) and concept drifts [7][9].

There are two approaches for stream classification: single model and ensemble model. A single model incrementally updates single classifier and known as incremental learning approach that deals with the classification task when datasets are too huge or when new examples can arrive at any time [15]. The combination of classifiers is used in ensemble model to create complex model, and handles concept drift very powerfully.

The rest of the paper is organized as follows: Recent techniques for data streams with drift are discussed under Section I. Section II briefs about Ensemble methods. Section III outlines the types, techniques and methods of concept drift and Section IV concludes the survey on concept drift in data stream classification using ensemble methods efficiently.

1. Related Work

Data growth has been tremendous and is considered useful only when all the data are processed efficiently. Data flow is in the form of data streams and it creates several new challenges for learning algorithms. In many real-time machine learning applications, drift can occur at any moment of time. It means the distribution of features and the labels tend to change over time that frequently affects the predictive performance of the model over time. A significant effort of recent research has focused on data stream classification tasks in non-stationary environments [4]. The foremost challenge in this research area concerns the adaptation to *concept drifts*, that is, when the data distribution changes over time in unpredicted ways and some of them are listed in this paper.

H. Ghomeshi et al., [5] proposed a new ensemble learning method that follows evolutionary algorithms to flawlessly adapt to concept drifts in non-stationary data stream classification. This method achieves the highest average accuracy and the best average rank among all methods. The overall evaluation time of the proposed method is higher and is considered as a limitation.

Abbaszadeh et. al., [1] proposed a classifier that employs the weighted majority voting to generate its final prediction. The drift is detected based on a new measure called "Kappa statistics" which emphasizes intelligence of the algorithm and adds a new perception to consider the quality of each classifier to drop base classifiers. The method has acceptable accuracy under gradual and combined drifts whereas with sudden drifts it degrades a little.

Sun et. al., [20] focused on a unique kind of concept drift called recurring concepts that appear after a certain period of time. A new method is introduced to handle this type of drift using classifier graph on the change detection method. It achieves better performance and robust against various noise levels and various types of drifts.

Yang et. al., [25] gives a new ensemble extreme learning machine that can be used in data stream classification tasks. This method adds a concept drift detection method and includes online sequence learning strategy in dealing with gradual concept drift and uses updating classifier in detecting abrupt concept drift thereby detection of abrupt and gradual drift is made possible.

Khammasi et. al., [12] proposed a new ensemble approach named Ensemble EDIST2 that combines three diversity techniques like block-based data, weighting data and filtering data. These techniques efficiently handle drift characteristics such as speed, severity that varies over time and acts as a drift detection mechanism to monitor ensemble's performance and detect changes. This method achieved the best accuracy rate and promotes stable behavior in all datasets.

Wanas et. al., [22] introduces the incremental knowledge concept drift (IKCD) algorithm. IKCD is a trigger based concept drift algorithm based on the data feature values in detecting concept drift. IKCD reduces the number of retrain required by the model and enhances the accuracy of the model.

The greater part of the above data stream learning approaches to non-stationary environments make use of ensemble learning techniques for classification tasks and in handling drifts.

2. Overview of Ensemble Methods

Ensemble is the process of adding various set of learners (unique models) as a whole for bringing out better stability and predictive power of the new model. An ensemble, also termed as multiple classifier or committee [2], is a collection of heterogeneous classifiers whose predictions are used to give new outcomes. A classifier is an algorithm that maps the input data to a specific category. An ensemble method maintains high predictive accuracy and drastically reduces training complexity. It is considered as the popular strategy for improving the predictive ability of a machine learning model. A good ensemble can be defined in such a way that where all the models combined to form a single model which is both accurate and diverse in their error. The model of a sample ensemble classifier is illustrated in Fig. 1.

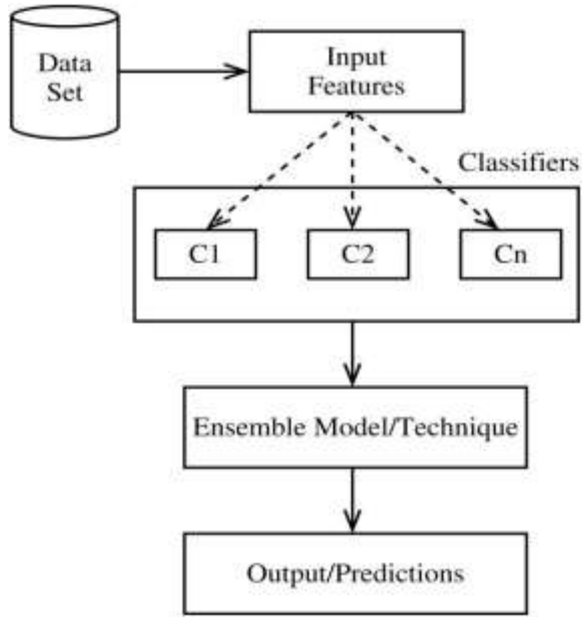

Fig 1: An Ensemble Classifier

Early contributions that pave the path to the scenario of ensemble methods are combining classifiers, ensembles of weak learners and mixture of experts. An ensemble contains a number of learners called base learners. They are generated from training data by a base learning algorithm like decision tree, neural network etc., and are mainly appealing as they convert weak learners to strong learners.

There are basically two ensemble paradigms based on how the base learners are generated.

1. **Sequential Ensembles**: The selection of base learners is done sequentially. It exploits the dependence between the base learners and the overall performance can be boosted.

2. **Parallel Ensembles**: The selection of base learners is done parallel. It exploits the independence between the base learners and reduces the error by the combination of independent base learners.

In recent years, ensembles of learners have been widely studied and deployed in real world problems. Dietterich provided three reasons that justify using ensembles provides effective results instead of single learners, that is, statistical, computational, and representational [26]. No free lunch Theorem formulated by Wolpert [26] mainly motivates the use of classifiers in ensemble methods. It states that “There is not a single classifier that meets the needs for all tasks, as every algorithm has its own uniqueness in its domain”.

2.1 Designing Approaches for Classifiers

The most important key factor for the ensemble is the selection of classifiers. Classifiers with high diversity and accuracy that are mutually complement is needed for designing an ideal ensemble. The accuracy of the ensemble is increased with classifier’s diversity and the following approaches are considered when designing classifier ensembles [2]:

- Coverage Optimization – generates mutually complementary classifiers designed together to achieve accuracy
- Decision optimization concentrates to design and train a decision combination function
- Recommending interconnections with individual classifiers in the ensemble
- Choosing more classifiers with diverse and complementary characteristics
- Suggesting a combination rule that shows the unique features of the component classifiers in deciding ensemble’s final role

Two simple strategies for combining ensemble classifiers are fusion and selection [17]. Ensembles using selection strategy finds the best classifier that is most capable of correctly classifying a particular instance and is known as **cooperative ensembles**. Ensembles using fusion methods use the outputs of selected classifiers to determine the label of an instance and referred to as **competitive ensembles**.

2.2 Types of Ensembles

The ensemble methods are basically categorized into two types [17]:

1. **Homogeneous Ensemble:** If the ensemble is built using same type of learning algorithm i.e., learners of the same type.
2. **Heterogeneous Ensemble:** If the ensemble is built using different type of learning algorithm, i. e., learners of different types. It is otherwise named as hybrid ensembles.

Heterogeneous ensemble will always give better accuracy than homogeneous ensembles as it uses different fine tune algorithms with the same datasets for each model that works with small amount of estimators and the latter do not use fine tune algorithms, uses different data sets for each model and works with large amount of estimators and it is also expensive.

2.3 Techniques in Ensemble Learning

There are some basic and advanced techniques like Averaging, Voting, Bagging, Boosting, Stacking are handled to enhance the overall performance of the classifiers combined. The Table 1 [27] lists the techniques of ensemble methods and its purpose with the specific algorithms.

Techniques	Purpose	Algorithm
Averaging	Popular and fundamental combination method for numeric outputs. Types: Simple and Weighted averaging	Multiple learning algorithms
Voting	Combination method used for classification problems. Types: Majority, plurality, weighted and soft voting	Multiple learning algorithms
Boosting	Converts weak learners to strong learners and decreases bias errors, homogeneous in nature	AdaBoost Algorithm
Bagging	Applies bootstrap sampling for training base learners and reduces variance, homogeneous in nature	Bagging Algorithm
Random Forest	Extension of Bagging with the incorporation of randomized feature selection	Random Tree Algorithm
Stacking	Train the learners (meta-learner) to combine individual learners (first-level learner), Heterogeneous in nature	Stacking Algorithm

Table 1 Ensemble Techniques

2.4 Errors in Ensemble Methods

The expected error [14] of a learning algorithm or a classifier can be broken down into three components:

1. A **Bias term** measures the accuracy of the average classifier obtained from the learning algorithm to that of the target function. The high bias term indicates that the model is under-performing.

2. A **variance term** measures the predictions of the learning algorithm different from each other. A high variance model indicates that it executes well on training data and performs bad beyond that data

3. A term for the minimum classification error is measured associated with Bayes optimal classifier for the target function and referred to as **Intrinsic noise or irreducible error**

Based on the above terms, the error in ensemble learning is given as

$$Error = (Bias)2 + Variance + IrreducibleError$$

For developing a good model, the developer need to find a good balance between the bias and variance as it minimizes the error rate gradually.

The three reasons [26] for which the traditional approaches fail and justify why ensembles are used instead of single learners are statistical, computational and representational issues. The learning algorithm is said to have **high variance** if it deals with statistical issue, a learning algorithm is said to have **high computational variance** if it deals with computational issue and a learning algorithm is said to have **high bias** when it suffers from representational issue. The combination of bias and variance error in any of the ensemble method can be reduced by applying the suitable ensemble techniques.

3. Concept Drift Framework

Concept drift represents that the primary or fundamental data changes its distribution over time and it is an observable fact in a data stream that refers to change in distribution of the data in the context of input data [10][23]. The data distribution changes over time and refers to the change in relationships between input and output data. It occurs due to the differences raised outside the scope of the data passed to the learning algorithm [7]. Concept means “target class” and concept drift means changes in the underlying distribution of the target classes [9]. Concept drifts affects the incoming instances and degrades the performance of the classifier [2].

To demonstrate the classification processes, Bayesian Decision Theory is commonly applied based on their prior probability distribution of classes,
i.e. p(y), and the class conditional probability distribution, i.e. p(X|y) [4]. The classification decision is associated to the posterior probabilities of the classes and is obtained by:

$$p(c_i|X) = p(c_i) \cdot p(X|c_i)/p(X)$$

where p(X) belongs to X, which is constant for all the classes of c_i [1]. The marginal probability distribution is given as [4]

$$p(X) = \sum {}_{(i=1)}^{m} p(c_i) \cdot (X|c_i)$$

Formally, a concept drift may occur when there is a change in the

- prior probability distribution p(y),
- class conditional probability distribution p(X|y), and
- posterior probability distribution p(y|X)

Data Streams that exhibits concept drifts are called as evolving or non-stationary data streams. Concept drift can also be termed as Covariate shift, population shift, and dataset shift [27].

3.1 Types of Concept Drifts

The important characteristic of concept drift deals with the rate at which it happens [7]. The types of concept drift are classified based on either time or the predictive views [16]. Based on time [18] there are four types of concept drift; namely (i) sudden, (ii) gradual, (iii) incremental, and (iv) recurring drift and given in Figure 2.

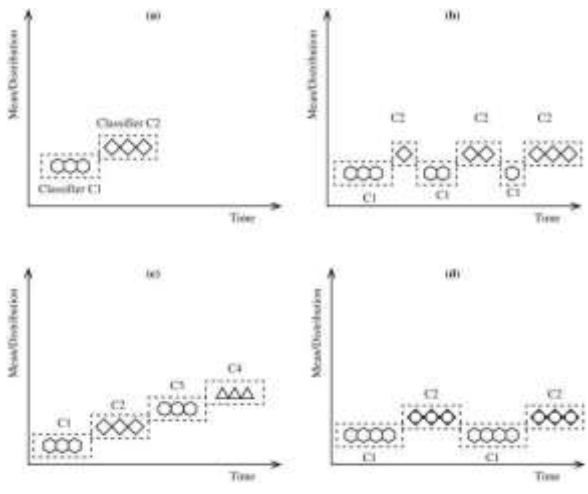

Fig 2: Types of drift based on time (a) Sudden (b) Gradual (c) Incremental (d) Recurring

(a). **Sudden Drift or Abrupt Drift** results from a sudden change in the data distribution. It takes place when a concept C1 is abruptly replaced by another concept C2. It means that the distribution of data will be changed directly to a new one in a relatively short time. It is easily recognizable for the reason that the prediction error and the data distribution differs much in a short span of time [2][7][8][16][24][27].

(b). **Incremental Drift or Stepwise drift** consists of a sequence of small changes. It can be identified only over an extended period of time, because small changes accumulate over time. It represents concepts that slowly evolve over time [7][22].

(c). **Gradual Drift** results from a slow transition from one data distribution to the next. That is, the two patterns may coexist concurrently. It is characterized by a transitioning window where instances from the new concept become predominant and instances from the previous concept are less frequent. It refers to the transition phase where the probability of sampling from the first distribution decreases while the probability of next distribution increases. It is connected with a slower rate of changes as data stream needs to be observed for a longer period of time and are not so radical [2][7][8][16][20][22].

(d). **Recurring Drift** refers to the case when a previously concept reappeared after some time or it happens whenever concepts keep recurring every so often or randomly. The recurrence of drifts could be cyclic [2][7][22].

The Figure 3 depicts two types of concept drift which are classified from the predictive point of view namely (i) Real drift and (ii) Virtual drift.

1. **Real Drift** refers to the transformation in the posterior probability of the distribution of the class membership p(y|x) and can also be referred as concept shift or conditional change.

2. **Virtual Drift** refers to the alteration in the data received i.e., change in the value of the attribute p(x) or class distributions p(y) does not affect decision boundaries. In other words, the evidence or the marginal distribution of the data p(x), changes without affecting the posterior probability of classes p(y|x).

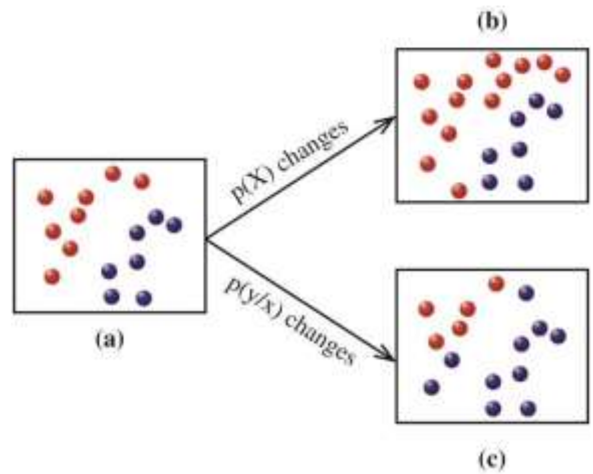

Fig 3: Types of drift from the predictive point of view (a) Original data (b) Virtual Drift (c) Real drift

3.2 Drift Handling Approaches

The major challenge for data stream classification and mining is handling concept drift that enables the update of the classification model. It uses techniques of the changed concept of the data to maintain accuracy [22]. To overcome the problem of concept drift, various learning algorithms have been proposed and ensemble methods have been considered as a best approach for learning a drifting concept in many cases [23]. Ensemble-based algorithms are considered to be more flexible in adapting concept drift.

Concept drift can also be handled by Adaptive learning that is implemented using Incremental and ensemble learning. Incremental learning follows machine learning paradigm and ensemble learning uses multiple base learners and combines predictions. Ensemble learning approach is the most popular evolving technique to handle concept drift [9].

The ensemble methods available are categorized into **explicit** and **implicit** methods. Explicit methods detects concept drift by selecting an appropriate concept drift detection mechanism and reflects explicit (immediate) reaction to a drift, while implicit methods does not provide an immediate reaction to concept drifts, and accepts to drifts in an implicit way by updating the ensemble status according to the most recent instances [5].

Ensemble based incremental learning algorithm uses two different methodologies to handle drifted data streams, namely **adaptive** and **wrapper** methods. Adaptive methods are incremental algorithms and are used for concept drift detection algorithms. It first identifies drift using novelty detection algorithms and adapts the classifiers to this change later. Wrapper methods are used for passive drift detection algorithm. The model is constructed based on the type of drift occurred and it has the capability to deal with recurring drifts as the construction of ensemble can be done parallel [21].

The other kind of approaches to cope with concept drift is **Active** and **Passive** approaches. Active approaches require change detection modules. It adapts a learner at regular intervals and updates the model without requiring explicit change detection. Passive approaches make adjustment whenever drift occurs and ensemble classifiers are one of the most popular passive approaches. Active approaches work well in handling sudden concept drift whereas passive approaches work better for gradual drift [20].

Ensembles of classifiers deal with time-changing streams naturally. The various methods to handle the drifts are illustrated in Figure 4.

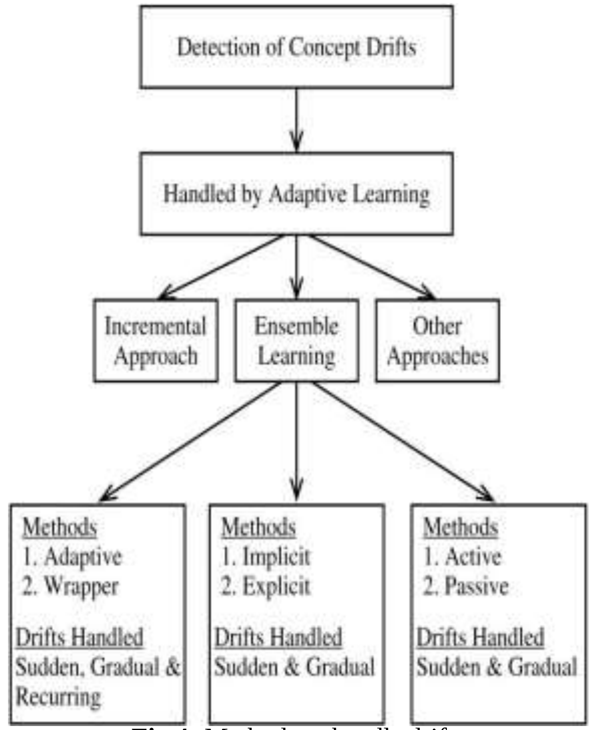

Fig 4: Methods to handle drifts

3.3 Drift Detection Methods

Detecting changes in data-streams is an important part of enhancing learning quality in dynamic environments [6]. The detection of concept drift allows pointing out when a data stream changes its behavior over time [3]. Concept drift detectors are methods that are based on the performance of the classifier or the data stream that signals the changes in distribution [2]. The detectors used to detect or monitor drifts are mainly classified based on Statistical Process Control, Sequential analysis, Contextual approaches and methods using two distinguished time windows for monitoring distributions of data.

(a). **Drift Detection Method (DDM)** monitors the classification error rate to detect change. It uses the concept of binomial distribution and when data distribution is stable, the model's error decreases with time and vice versa [1]. It takes two modes like **warning level** in which the input data is stored in temporary memory and **drift level** in which the model algorithm works by relearning from the data in temporary memory [2]. DDM works best on data streams with sudden drift as gradually changing concepts can pass without triggering the level of the drift [4]. DDM learns from online classifier and keeps track of error rate.

(b). **Early Drift Detection Method (EDDM)** is a revision of DDM to advance the detection of gradual drifts and its key idea is to consider the time interval (distance) between two occurrences of classification errors. They assume that any significant decrease in the distance suggests that the concept is changing. EDDM performs well for gradual changes; however, it is not good at detecting drift in noisy examples. EDDM stores previous algorithm in short term memory and it shows higher performance consuming less storage [1][2][13].

(c). **Adapting Sliding Window Algorithm (ADWIN)** is the best acknowledged method in comparing two sliding windows that has a variable size. The average value of the two windows is calculated and compared. The recent sub-window data will be dropped if it exceeds the acceptable level. ADWIN works only for one-dimensional data. A separate window must be maintained for each dimension, for n-dimensional raw data, which results in handling more than one window. It uses two windows: (i) a reference window, and (ii) a test window. The reference window helps to calculate the error rate and other parameters when there is no change and the output value is taken as a reference. The test window represents the new data batch and the algorithm monitors its error rate in detecting the concept drift. A threshold value is declared to identify the existence of the drift, based on the average error rate of the reference sub-windows and other parameters [2][5][11][22].

(d) **Adaptive Boosting (Aboost)** is one approach that builds classifier that classifies the instances from data stream which receives block of data and uses concept drift methods. Then, the output of the ensemble is evaluated and the weights of all classifiers are updated when an instance is classified correctly by the ensemble, as well as by the classifier itself. If a concept drift is detected, the ensemble resets the classifier's weight to one and the oldest classifier is removed from the ensemble whenever the size exceeds. [5]

(e) **Dynamic Weighted Majority (DWM)** is an implicit approach and the selected classifiers gains knowledge using incoming samples from the stream of data in an incremental approach. It is a representative method using weighted majority voting. It maintains a couple of learners trained in different datasets in different time periods where each learner has a weight to specify how reliable it is. All the weights are updated over time according to the evaluation of the new datasets and the learners with low weights are removed or replaced with new learners. The overall system makes predictions using weighted majority voting among the base learners. One advantage of DWM is the total base learners that should be used is initially specified and this set of base learners is continuously updated by the training process to reflect this [5][24].

(f) **The Accuracy Updated Ensemble (AUE) algorithm** is an extension of Accuracy Weighted Ensemble and trains all existing classifiers one by one and assigns weight based on error occurrence in a constant time and memory. The algorithm combines the incremental nature of Hoeffding trees with a normal block-based weighting mechanism. The removal of old classifiers is not performed and assigns a threshold for memory so that whenever it is met, a pruning method is applied for reducing the size of classifiers [5][21].

(g) **Streaming Ensemble Algorithm (SEA)** is a pioneering method for dealing with drift detection in streaming data. It uses ensemble classifier approach and maintains a constant number of classifiers in its ensemble pool. When a new dataset is available, it performs majority classification on the new instances. It then reevaluates the composite classifiers according to their classification accuracies and replaces some underperforming classifiers with new classifiers. The overall accuracy is improved by using the updated classifiers. It uses a simple majority vote and do not perform well in recurring environments [21][24].

Drift Detection Methods	**Drift detected**	**Methodology**
Drift Detection Method (DDM)	Sudden Drift	Computes error rate of the base learner
Early Drift Detection Method (EDDM)	Gradual Drift	Computes distance error rate of the base learner
Adapting Sliding Window Algorithm (ADWIN)	Sudden Drift	Variable size adaptive sliding window method
Adaptive Boosting (Aboost)	Gradual Drift	Reset the value of classifier when the drift is detected
Dynamic Weighted Majority (DWM)	Gradual Drift	Based on weighted majority algorithm
The Accuracy Updated Ensemble (AUE)	Gradual Drift	Enhancement of AWE and uses incremental classifiers and uses Hoeffding tree as classifiers
Streaming Ensemble Algorithm (SEA)	Gradual Drift	Uses unweighted majority voting and heuristic replacement strategy

Table 2 Drift Detection Methods

The Table 2 summarizes the drift detection methods and the type of drift detected in each method.

Conclusion

An Ensemble method creates awareness and interest in the machine learning community. Techniques such as Boosting, Bagging and Stacking are so effective to prove their performance in comparing single learners with combined learners. Ensembles need to properly optimized and important in understanding the mechanisms through which it achieves effectiveness. Handling concept drift is becoming an attractive topic of research that concerns multidisciplinary domains such that machine learning, data mining, ubiquitous knowledge discovery, statistic decision theory, etc... The concept drift problem is very broad and new ideas are being continuously developed over time. Concept drift learning in data streams can be dealt efficiently with ensemble methods. Various learning algorithms have been proposed to tackle the concept drift inherent in data stream and ensemble methods have been verified as a best approach for learning a drifting concept in many cases. The goal of this survey is to present an effortless perception of the concept drift issues and associated works, in order to help researchers from different disciplines to consider concept drift handling in their applications. This paper covers diverse components of existing approaches, evokes discussion and helps readers to underline the sharp criteria that allow them to properly design their own approach as the existing state-of-the-art is presented with advantages and limitations.

References

[1] Abbaszadeh, O., Amiri, A. & Khanteymoori, A.R. Frontiers Inf Technol Electronic Eng (2015) 16: 1059. https://doi.org/10.1631/FITEE.1400398

[2] Bartosz Kraawczyk, Leandro L. Minku, Joo Gama, Jerzy Stefanowski, and Micha Woniak. 2017. Ensemble Learning for data stream analysis. Inf. Fusion 37, C (September 2017), 132-156. DOI:https://doi.org/10.1016/j.inffus.2017.02.004

[3] da Costa, F. G., & de Mello, R. F. (2014, October). A Stable and Online Approach to Detect Concept Drift in Data Streams.In *2014 Brazilian Conference on Intelligent Systems* (pp. 330- 335).IEEE.

[4] Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., &Bouchachia, A. (2014).A survey on concept drift

adaptation. *ACM computing surveys (CSUR)*, *46*(4), 44.

[5] Ghomeshi, H., Gaber, M. M., &Kovalchuk, Y. (2019). EACD: evolutionary adaptation to concept drifts in data streams. *Data Mining and Knowledge Discovery*, *33*(3), 663-694.

[6] Harel, M., Mannor, S., El-Yaniv, R., & Crammer, K. (2014, January).Concept drift detection through resampling. In *International Conference on Machine Learning* (pp. 1009-1017).

[7] Heitor Murilo Gomes,Jean Paul Barddal, Fabrício Enembreck, & Albert Bifet. 2017. A Survey on Ensemble Learning for Data Stream Classification. ACM Comput. Surv. 50, 2, Article 23 (March 2017), 36 pages. DOI:https://doi.org/10.1145/3054925

[8] Hewahi, N. M., & Kohail, S. N. (2013). Learning Concept Drift Using Adaptive Training Set Formation Strategy. *International Journal of Technology Diffusion (IJTD)*, *4*(1), 33-55.

[9] Janardan, Shikha Mehta, Concept drift in Streaming Data Classification: Algorithms, Platforms and Issues, Procedia Computer Science, Volume 122, 2017, Pages 804-811, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2017.11.440.

[10] Joao Gama. 2010. Knowledge Discovery from Data Streams (1st. ed.). Chapman & Hall/CRC.

[11] Kadwe, Y., & Suryawanshi, V. (2015).A review on concept drift. *IOSR J. Comput. Eng*, *17*(1), 20-26.

[12] Khamassi, I., Sayed-Mouchaweh, M., Hammami, M., &Ghédira, K. (2019).A New Combination of Diversity Techniques in Ensemble Classifiers for Handling Complex Concept Drift. In *Learning from Data Streams in Evolving Environments* (pp. 39-61). Springer, Cham.

[13] Nishida, K., & Yamauchi, K. (2007, October).Detecting concept drift using statistical testing.In *International conference on discovery science* (pp. 264-269).Springer, Berlin, Heidelberg.

[14] Opitz, D., & Maclin, R. (1999). Popular ensemble methods: An empirical study. *Journal of artificial intelligence research*, *11*, 169-198.

[15] Patel, M., & Patel, Y.M. (2015).Detection of Novel Class for Data Streams.

[16] Pesaranghader, A., Viktor, H. L., &Paquet, E. (2018, July). McDiarmid drift detection methods for evolving data streams. In *2018 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-9). IEEE.

[17] Polikar R. (2012) Ensemble Learning. In: Zhang C., Ma Y. (eds) Ensemble Machine Learning. Springer, Boston, MA

[18] Sidhu, P., & Bhatia, M. P. S. (2015). Empirical support for concept drifting approaches: Results based on new performance metrics. *International Journal of Intelligent Systems and Applications*, *7*(6), 1.

[19] Sumathi, M. R., & Poorna, B. (2017). Design and Development of Ensemble of Naïve Bayes Classifiers to predict Social and Communication Deficiency among Children. *International Journal of Applied Engineering Research*, *12*(24), 14190-14198.

[20] Sun, Y., Wang, Z., Bai, Y., Dai, H., & Nahavandi, S. (2018). A Classifier Graph Based Recurring Concept Detection and Prediction Approach. *Computational intelligence and neuroscience*, *2018*.

[21] Thalor, Meenakshi & Patil, S. T. Patil. (2016). Incremental Learning on Non-stationary Data Stream using Ensemble Approach. International Journal of Electrical and Computer Engineering (IJECE). 6. 1811-1817. 10.11591/ijece.v6i4.10255.

[22] Wanas, N., Farouk, A., Said, D., Khodeir, N., &Fayek, M. (2019). Detection and Handling of Different Types of Concept Drift in News Recommendation Systems. *International Journal of Computer Science & Information Technology (IJCSIT)*, *11*(1), 87.

[23] Wang, L. Y., Park, C., Choi, H., & Yeon, K. (2016).A classifier ensemble for concept drift using a constrained penalized regression combiner. *Procedia Computer Science*, *91*, 252-259.

[24] Wang, R., Shi, L., O Foghlu, M., & Robson, E. (2010).A Meta-Learning Method for Concept Drift.

[25] Yang, R., Xu, S., &Feng, L. (2018).An ensemble extreme learning machine for data stream classification. *Algorithms*, *11*(7), 107.

[26] Zhi-Hua Zhou. 2012. Ensemble Methods: Foundations and Algorithms (1st. ed.). Chapman & Hall/CRC.

[27] Žliobaitė, I., Pechenizkiy, M., & Gama, J. (2016).An overview of concept drift applications. In *Big data analysis: new algorithms for a new society* (pp. 91-114). Springer, Cham.