EnsembleConceptDrift Detection in Data Stream Mining: A Review

Hadi Tarazodar[1],KaramollahBagherifard[2], Samad Nejatian[3], Hamid Parvin [4], Razieh Malekhosseini[5]
[1,2,5]Department of computer Engineering ,Yas.C., Islamic Azad University ,Yasuj, Iran
[3]Department of ElectricalEngineering ,Yas.C., Islamic Azad University ,Yasuj, Iran
[4]Department of computer Engineering ,NoM.C., Islamic Azad University ,Noorabad Mamasani, Iran
Emails: [1]Hadi.Tarazodar@iau.ac.ir , [2]Ka.bagherifard@iau.ac.ir(corresponding author) ,
[3]Sa.nejatian@iau.ac.ir , [4]Parvin@iaut.ac.ir , [5]Malekhoseini.r@iau.ac.ir
Receive Date: 07 July 2025 Revise Date: 27 July 2025 Accept Date: 29 July 2025

Abstract

In the data-driven era, machine learning plays a vital role in analyzing and processing big data. One of the fundamental challenges in this area is managing conceptual drift in data streams, where the changing distribution of data reduces the accuracy of learning models and makes them ineffective in predicting the future . Traditional classifiers are not expected to learn patterns in non-stationary distributions of data. For any real-time use, the classifier must detect concept drift and adapt over time.Compared with concept drift detection for a data stream, the challenges of ensemble concept drift detection arise from three aspects: first, the training data becomes more complex, Second, the underlying distribution becomes more complex , and third, the correlation between data streams becomes more complex . In this article, we provide a comprehensive review of ensemble concept drift detectors in data stream mining, and also review their techniques, key points, advantages, and limitations .

Keywords : Data stream mining, concept drift, concept drift detection, ensemble learning

1- Introduction

Today, many organizations are continuously generating huge amounts of data, with a much greater speed and volume than ever before. For example, Google conducts over 3.5 billion searches daily, NASA satellites generate about 4 terabytes of imagery, and Walmart records over 20 million transactions . This data is so large that it is not stored in main memory and is instead stored on secondary storage devices. As a result, random access to this data, which is assumed in many traditional data mining algorithms, is very expensive and time-consuming [1].

Streaming data is defined as “an unlimited sequence of multidimensional, sparse, and transient observations that are available over time” [2]. In other words, streaming data consists of a sequence of samples of the form $\{x_1, x_2, ..., x_n\}$, where x_1 is the first sample and $x_n$is The last sample has been imported. Each sample x_i is an n- dimensional feature vector consisting of features $A_i = \{A_1, A_2, ..., A_n\}$ with a class label C_i . For example, the training data is represented as $\{x_1, x_2, ..., c\}$ where x_i are the samples and c are their classes.

The unlimited and dynamic nature of data streams creates certain technical and operational limitations that make traditional data streaming algorithms face serious challenges due to the high resource consumption (time, memory, and processing) for processing dynamic and evolving data distributions. These challenges are especially evident in cases where the data is constantly changing and growing. Therefore, one of the key issues in this field is the design of algorithms that are capable of processing data in real time or near real time with optimal use of limited resources. These algorithms must be designed in such a way that they can automatically adapt to changes in the data stream and have high efficiency and accuracy .

One of the major challenges in learning from streaming data is a phenomenon called concept drift, which refers to changes in the distribution of data over time. In this phenomenon, the underlying relationships and patterns of the data may change gradually or suddenly, posing a major challenge to traditional machine learning models that assume a fixed distribution of data. These changes can be caused by factors such as equipment failure, intrusion, seasonal changes, or even changes in consumer behavior [3].

Ensemble learning refers to the combination of multiple learning models to solve a specific problem. This method uses the combination of multiple models' predictions to increase accuracy and reduce prediction error. In ensemble learning, different learning models are used to predict an outcome and then their predictions are combined to achieve a better prediction [4].

One of the review articles related to this research is the article [5], which provides a general classification of concept drift detection methods up to 2020, focusing on classical algorithms in supervised and semi-supervised learning environments.

However, the aforementioned article does not address recent developments and advances, especially in the field of ensemble learning algorithms that have been proposed in recent years. In contrast, the present article, by providing a comprehensive review of ensemble learning-based concept drift detection methods up to 2025, covers the gaps in previous studies and, in addition, provides a comprehensive comparative analysis of the techniques used, key points, advantages and limitations of each method.

The paper is organized as follows: Section 2 discusses ensemble learning. Section 3 discusses the concept of drift and its types. The existing ensemble drift detection algorithms are shown in Section 4. The conclusion is in Section 5 and future work is discussed in Section 6.

2- Ensemble learning

Ensemble learning is a technique used to combine two or more algorithms. Machine learning is used to achieve superior performance compared to when the constructive algorithms are used individually . Instead of relying on a single model, the learners' predictions are combined using a combination rule to obtain a single prediction that is more accurate. The general framework of any ensemble learning system is that it uses an aggregation function G to combine a set of base classifiers to predict a single output . Given a dataset of size n and features of dimension m ,$D = \{x_i . y_i\}. 1 \le i \le n . x_i \in R^m$ the output prediction based on this ensemble method is expressed by Equation 1. Figure 1 shows the abstract general framework of ensemble learning.

$$y_i = \phi(x_i) = G(c_1 . c_2 . \dots . c_k) \qquad (1)$$

In general, ensemble methods can be classified into parallel and sequential ensembles . Parallel methods train different base classifiers independently and combine their predictions using a combiner. A common parallel ensemble method is bagging and its extension, the random forest algorithm [6]. Parallel ensemble algorithms use parallel generation of base learners to encourage diversity in ensemble members.

Meanwhile, sequential ensembles are not independently fit to the baseline models . They are trained iteratively so that the models learn to correct the errors of the previous model at each iteration . A popular variant of sequential ensembles is the boosting algorithm [7]. Furthermore, parallel ensembles can be classified as homogeneous or heterogeneous, depending on the baseline learners. Homogeneous Homogeneous ensembles include models that are generated using the algorithm Machine learning They are made the same.

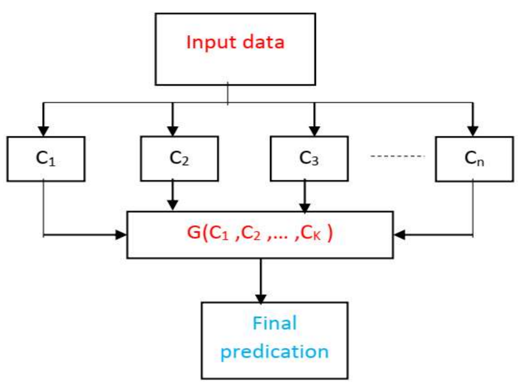

Fig.1. General framework for ensemble learning

3- Concept drift

In a continuous data stream, suppose that the distribution of the data changes over time.At any time t, the distribution of the data exists as $p(X_t|C_t)$, which models the probability of observing data $X_t$in class C_t.Conceptual drift refers to changes in the distribution of data over time. Mathematically, conceptual drift occurs when the distributions $p(X|C)$change over time, such that the distribution of the past data $p(X|C)$can no longer accurately model the new data X_{t+1}.In this case, if for each time t, the data distribution is $p(X_t|C_t)$, then conceptual drift can be modeled as changes in these distributions as follows:

$$p(X_t|C_t) \neq p(X_{t+1}|C_{t+1}) \qquad (2)$$

These changes in the data distribution must be continuously detected by the learning model.

Types of drifts

Data streams are continuous and the distribution of real -time data is non-stationary. The distribution of data may vary over time. These changes in data , namely real concept drift and virtual concept drift , can be considered as two types of drift. Figure 2 describes the types of drift in terms of speed .

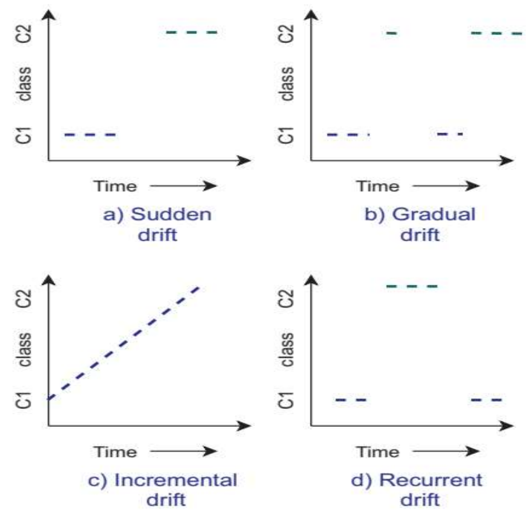

Fig.2. Types of driftsin terms of speedDrift

Sudden drift

Here , the new concept of the incoming data stream suddenly replaces the old concept .Therefore , the point in time when the old concept suddenly changes to the new concept is known as the sudden drift (SeeFigure 2(a)) .

Gradual drift

In gradual drift, the duration of the concept change is relatively long compared to sudden drift (see Figure 2(b)). There are two types of variations in this type of drift: slow gradual drift and normal gradual drift.

Repetitive drift

In this type of drift, the concept reappears after a long period of time ,i.e. , a repeated change in the concept occurs in the flow (See Figure 2(d)). It has cyclical and non - cyclical behavior . Cyclical phenomena apply in conditions where seasonal changes occur .

Incremental drift

In incremental drift, an old concept gradually transforms into a new concept over a period of time (see Figure 2(c)).

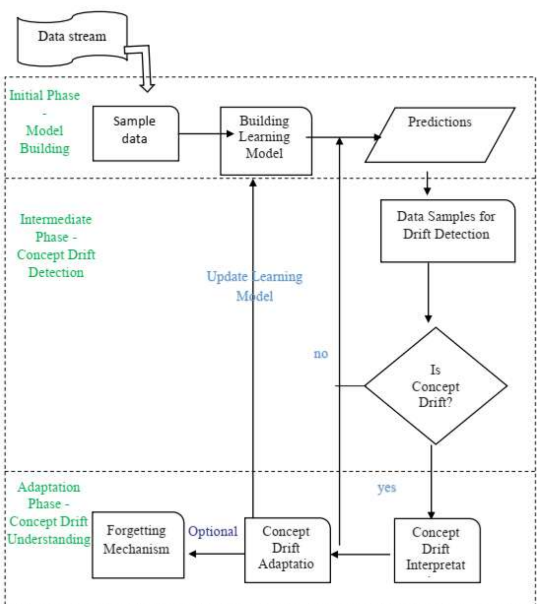

Fig.3. General block diagram of concept drift detection

4- Concept drift detectors

This concept is not stable because it changes over time . These changes make the model unadaptable ;therefore , it is necessary to update a model regularly. Changes in the time -related distribution may cause errors . A learning model is augmented .Therefore , the error detection mechanism tracks the errors online . In this paper , concept drift detection algorithms are divided into several categories . Figure 3 The general block diagram of drift detection illustrates the concept .

4-1-Methods based on ensemble learning

Most ensemble-based discriminators are based on the Weighted Majority Algorithm (WMA) method [8]. They are made. WMA selects the best learners by giving weight to each of them based on their performance .

The Stream Ensemble Algorithm (SEA) approach by Stream and Kim [9] is a conceptual approach to deal with drift. SEA implicitly manages this drift by creating a new learner for each new piece of data until the maximum number of learners is reached . Learners are refined based on their prediction performance . It uses majority voting to map the output predictions of classifiers to the ensemble predictions . It is not best suited to deal with late or missing labels because the ensemble relies on prior accuracy (and thus on timely correct class labels) to replace the lowest quality classifier.

A similar approach to ensemble refinement was introduced in the accuracy weighted ensemble (AWE) by Wang et al[10]. The idea of AWE is to weight each classifier using a specific type of mean square error in the most recent chunk using cross-validation. The weight of a classifier is inversely proportional to its prediction error estimate. Classes are pruned if they predict poorly or worse by chance, or by only having a subset of those with the highest weights. This eliminates classes that do not model the current concept well or makes it difficult to create new classes to learn new concepts . However, just like previous classifiers, AWE adheres to the previous accuracy for its pruning strategy and therefore may have problems with missing or late-arriving class labels and requires a different pruning strategy to extend it to deal with semi-supervised data.

Proposed Adaptive Boosting (Aboost) ensemble , which combines Boosting using a chunk-based input . To detect concept drift, each time a chunk is received, the error of this ensemble is calculated. If a concept drift is detected, the entire ensemble is completely reset; otherwise, each instance of the chunk is assigned a weight based on the ensemble error. This weight is then used to train a new classifier from the weighted chunks, which are added to the ensemble if they are incomplete. Otherwise, the oldest classifier in the ensemble is replaced . Soft voting is used to map the classifier's prediction to a single output for the ensemble. In an experimental evaluation, they found that their approach outperforms SEA and AWE in terms of predicted accuracy . Their technique is also faster , uses less memory, and is more adaptive to concept drift.

Dynamic Weighted Majority (DWM) [11] which uses a weighting mechanism inspired by WMA . Each learner's weight is reduced by a multiplicative factor β , $0 \leq \beta \leq 1$, when it makes an incorrect prediction at each time step ρ.

One well-known ensemble-based drift detection method that has a chunk-based approach inspired by boosting is Learn++.NSE (Incremental Learning for NSEs) proposed by Elwell and Pallikar[12]. Learn++.NSE It is for dealing with unstable environments .In Learn++.NSE , a set of learners is trained on chunks of data examples . The training samples are weighted according to the ensemble error in these samples . The sigmoid function is used to weight learners in the ensemble based on their errors in the old and current parts .

Superfast forest of binary trees (UFFT) [13] is created with a ensemble of halving trees . The partitioning criterion used can only be applied to binary classification problems, but binary decomposition allows multi- class problems to be considered as well . Each pair of classes has its own binary tree, which is updated when a new instance has a true class label for one of the two classes .

Nishida and Yamauchi [14] Advanced version of adaptive taxonomic ensembles (ACE) proposes an algorithm that adds a pruning method and improves the voting method . ACE consists of an online classifier, a set of classifiers , and a drift detection mechanism. The online classifier is trained on each input sample and a fixed buffer that holds the most recently seen samples . When the buffer is full or a change is detected, a new classifier is created to summarize the data for that time period , the buffer is flushed, and the online classifier is retrained. A weighted majority vote is used to calculate the output for this ensemble.

Buffett et al. [15] propose Adaptive Window Bagging (ADWIN) which is only the result of a drift detector. ADWIN for bagging It is online. ADWIN is responsible for replacing the worst classifier in the ensemble with a new classifier when a change is detected . Others can learn using online ensembles on non -stationary streams , including the SAND semi-supervised framework.

Ditzler [16] proposed a framework that includes two related ensemble-based approaches, namely Learn++.CDS and Learn++.NIE . They extended their previous work in Learn++.NSE to accommodate unbalanced class data. These methods monitor the performance of both the majority and minority classes .Learn++.NSE (Non-stationary and Non-equilibrium Environments) and Learn++. CDS (Conceptual Drift with SMOTE) are introduced as two new members of the Learn++ family of incremental learning algorithms that explicitly and simultaneously address the aforementioned phenomena . The first addresses concept drift and class imbalance through modified packing-based sampling and replacing the class-independent error weighting mechanism—which typically favors the majority class—with a set of measures that emphasize good predictive accuracy across all classes.

Diversity to counter drift (DDD) [17] controls the level of diversity of learners in the ensemble by combining low-diversity and high-diversity ensembles . The low diversity ensemble is used for drift detection and the high diversity ensemble is used after drift detection .

Brzezinski and Stefanowski [18] proposed the Accuracy Updated Ensemble (AUE) algorithm, which improves AWE by conditionally updating component learners instead of adjusting weights . The authors also used a simpler weighting function than AWE .

Parameter-insensitive ensemble prediction (PINE) [19]is an ensemble approach that processes asynchronous concept classification in distributed networks . A modified version of the ADWIN drift detector is provided for each counterpart of the framework. The detector It monitors a stream of precision represented by ones and zeros .

Elderly weight ensemble (WAE) proposed by Woznik et al. [20] is inspired by AWE and generalized with two modifications. The first is that the classifiers are weighted based on prior accuracy as well as on how much time is spent within the ensemble . The second is that the latest modification adds classifiers to the ensemble based on their size of diversity.

AUE2 [21] improved AUE by introducing weighting and cost - effective pruning of learners. Updated Online Accuracy Unit (OAUE) [22] It uses a drift detector built into an online learner to generate a reweighting signal to the learner. The updated accuracy and growth rate (AGE) [23] ensemble extends AUE2 to respond to different types of drift. AGE uses the geometric mean to design the growth rate of basic learners .

DDE [24] built a small ensemble to control how the three drift detectors work and block their signals at the warning level and the drift level. Depending on how sensitive the DDE is, it requires a certain number of detectors to confirm the warning level or the drift level . Another parameter is the type of drift mechanism used . But each sensitivity setting has a default detector set that goes with it.

Online Weighted Ensemble (OWE) [25] It was proposed to adapt Learn++ for regression tasks, which can progressively learn an example in the presence of multiple types of changes and simultaneously preserve old information in recurring scenarios . The key idea is to keep a floating window that slides when a new instance is available. The error of each model in the current window is determined using a boosting strategy that assigns small errors to models that accurately predict poorly predicted samples from the ensemble .

Diverse Online Ensembles Detector (DOED) [26]It maintains two sets of weighted ensembles: one with high diversity and one with low diversity . The algorithm is based on comparing these two accuracies: the accuracy on recent data samples and the accuracy from the beginning of learning . It develops two ensembles with different levels of diversity E0 and E1 . DOED uses only one significance level to detect conceptual drift with E0 and E1 using P-value . Other category benchmark methods, statistical process control and windowing techniques, were also used in ensemble frameworks.

Lee et al. [27] used ensemble decision trees for conceptual drift. (EDTC) introduces a type of random feature selection where species perform split tests and use two Hafding boundary inequalities with specified thresholds. Random feature selection is performed instead of deliberate split tests . It creates a random ensemble that is incremental and based on a random decision tree. It dynamically adjusts the drift checkpoint and window size to detect conceptual drift .

ELM has also been employed in a ensemble approach to combat conceptual drift. An ensemble of online sequential extreme learning machines (ESOS-ELM)[28] was proposed to deal with conceptual drift in class imbalance data. ESOS-ELM maintains a ensemble of OS-ELMs and monitors the error rate using a threshold-based technique .

To overcome the drawbacks of DWM , which does not consider learner performance in the training data , DWM-WIN It was suggested in [29]. DWM-WIN is a ensemble method that incorporates the learner's age into the weighting mechanism and tracks conceptual drift in the learning phase .

Number and distance of errors (NDE) [30] It is a ensemble method that detects conceptual drift based on the number and distance between errors and compares it to a threshold .

Efficient control of concept drift and concept evolution on streaming data (ECHO) [31] It is a ensemble-based semi-supervised framework that includes a conceptual drift detection technique. ECHO keeps a sliding window on the data stream to monitor significant changes in the classifier's confidence to detect concept drift using the CUSUM test.

Gomez et al. [32] Adaptive Random Forest (ARF) proposed a method for classifying evolving data streams, which includes an efficient resampling method and adaptive operators that can deal with different types of conceptual drift without complex optimization for different data sets.

Knowledge Maximum Ensemble (KME) [33] is a concept drift detection system that contains a concept drift detector. which checks whether the ensemble classification error falls within a sliding window under the confidence interval.

Recursive Dynamic Weighted Majority (RDWM) [34] It is based on DWM by forming two ensembles of learners. The primary ensemble represents current concepts , and the secondary ensemble consists of the most accurate learners.

Heterogeneous Dynamic Weighted Majority (HDWM) [35] It was proposed to transform DWM into a heterogeneous ensemble by automatically selecting the best learners for use over time to prevent performance degradation.

Da Silva et al. [36] A Stacked method , called Fast and Deep Stacked Network (FDSN) suggested that it deals with static data sets. The authors suggested using multiple small SLFNs instead of using one large SLFN . Using an ELM -based algorithm to train all modules, FDSN has achieved similar performances (average error) on regression tasks compared to a large SLFN , while spending less time in its training phase and using much less memory than the compared methods.

Repetitive Adaptive Classifier(RACE) ensemble [37] maintains an archive of diverse learners and uses EDDM to detect repetitive drifts . Online drift detector for k-class problem (ODDK) [38] It was proposed to handle multi- class problems with conceptual drift. This algorithm builds a contingency table that stores the variation of a pair of classifiers and uses the PH test to detect conceptual drift .

Komornichak et al. [39] Statistical Drift Detection Ensemble (SDDE) proposes a new method for detecting conceptual drift . This method uses drift measures and conditional marginal variable drift measures that are analyzed by a set of discriminators , whose members focus on random subspaces of flow characteristics .

A method for detecting ensemble drift (GDDM) for multiple data streams was introduced by Yu et al. [40].The idea of the method is inherited from the error rate-based drift detection method for a data stream, i.e., the error rate is the input variable of GDDM instead of the data itself to ignore the differences in the number and scale of features. Instead, the difference is that the input variables in GDDM are multivariate because the error rates of all data streams are considered simultaneously. In addition, it has introduced a new test statistic to ignore the underlying distribution of data streams and the correlation of data streams.

A semi-regulatory framework called CPSSDS It was introduced by Tanha et al. [41] which uses an incremental classifier as the base learner and a self-learning framework to handle the shortage of labeled examples. In this The approach uses a form of matched predictors to discover a set of unlabeled learner data samples to add to the main training set in any training method, which is the main challenge in the standard self-learning approach. The Kolmogorov-Smirnov test is adopted to detect concept drift by comparing the coherent prediction outputs for two sequences of data chunks.

A supervised online method based on a class called the fast and deep sequential online stack network. (OSFDSN) was introduced by DaSilva and Ciarelli [42]. In this methodFast Deep Stack Network (FDSN) as a ensemble of Single-layer feedforward neural networks (SLFNs) are considered, where the output of the network is the output of the most recent SLFN . Online sequential FDSN (OSFDSN) is similar to FDSN , but each of its SLFN modules has a weighted contribution to the network output . These weights are calculated dynamically and based on the latest data .

Another method is the conceptual drift detection model based on Bidirectional Temporal Convolutional Network and Multi-Stacking Ensemble Learning (CD-BTMSE)[43]. CD-BTMSE selects six suitable base learners to solve the overproblems, poor generalization ability, and poor robustness of ensemble learning-based conceptual drift detection models, also using the bidirectional temporal convolutional network model.

(BiTCN) to improve detection accuracy. Concept drift by considering the temporal characteristics of the data as well as the two-way semantics in the recognition process . At the same time, it uses the Multi-Stacking ensemble learning model to solve the problem of low accuracy of concept drift detection caused by the relatively high generalization error rate and poor generalization ability of existing ensemble learning-based methods.

Aurora et al. [44] proposed a new approach – Selective Ensemble Using Transfer Learning (SETL) – that has the ability to accommodate the new concept of data . This approach uses transfer learning and a weighted majority voting scheme to optimize resources. It also overcomes issues such as negative transfer and overfitting that may occur during the transfer learning process .

Deng et al. [45] (proposed a new ensemble learning model called In-sample Weighted Ensemble Learning with Tripartite Decision-Based Example (IWE-TWD) . In IWE-TWD , a divide-and-conquer strategy is used to manage uncertain drift and select base learners. Density clustering dynamically constructs density regions to lock in the drift range. A three-way decision is made to estimate whether the area distribution changes, and the sample is weighted by the probability of the area distribution changing . The variation among basic learners is also determined by a tripartite decision.
For detail refer to Table 1. in appendix.

5- Conclusion

Drift , novelty detection, infinite length data streams, etc. are the main challenges in the streaming environment. Many ensemble drift detectors have been developed to detect concept drift. Most of the detectors are based on posterior distribution, error rate variation, threshold, etc. Conceptual drift detection methods are a ensemble of problems that suffer from many performance factors. These factors include slow adaptation to drift, poor sensitivity to drift types, high false positive rate, high computational complexity, and delay in detecting different drift types. The need for precise parameter adjustment , Dependence on model quality ,novelty detection, and detection of only some concept drift are other major concerns in data stream mining. New methods based on deep learning and the three-state decision framework contribute significantly to ensemble learning in detecting implicit drift .

In this article, we have conducted a complete and comprehensive review of ensemble concept drift detectors in data stream mining, and in addition, we have examined their techniques, key points, advantages, and limitations . We have also examined the types of sudden and gradual, incremental, and iterative drifts performed in these ensemble concept drift detectors .

Future works:

Future research could be directed in this direction:

1-Another review Drift detector for collaborative learning and evolving fuzzy systems, etc.

2-Comparing ensemble drift detectors in terms of homogeneity and heterogeneity and examining their evaluation criteria

3-Comparing ensemble drift detectors in terms of parallel and sequentiality and examining their evaluation criteria

Table 2. Review of techniques used in ensemble drift detection methods

Tekken Ones	Algorithm
learning -ensemble learning	Stream Ensemble Algorithm (SEA)
Ensemble learning - accuracy weighted ensemble	Accuracy Weight Ensemble (AWE)
Ensemble learning - Online learning - And performance-based marriage	Dynamic Weighted Majority (DWM)
Ensemble Learning - Incremental Learning - Learn++ - Weighted majority voting based on dynamic error	Learn++.NSE
Ensemble Learning - Incremental Learning - Learn++ - SMOTE - Weighting	Learn++.CDS
Ensemble Learning - Incremental Learning - Learn++ - Weighting	Learn++.NIE
Ensemble learning - online learning - combining variety and weighting	Diversity to deal with drift (DDD)
Online learning - Ensemble learning - Mean square error (MSE) - Weighting	Updated Accuracy Unit (AUE)
Ensemble learning –ADWIN -Asynchronous classification	Predictive Parameter Insensitive Ensemble (PINE)
Ensemble learning - time-based weighting - combining diversity and accuracy	Weighted Elderly Ensemble (WAE)
Online learning – ensemble learning –Weighting function based on incremental error	Online Accuracy Updated Ensemble (OAUE)
Ensemble learning - Incremental learning - Regression models -boosting	Online Weightlifting (OWE)
Ensemble learning - online learning - based onaccuracy and diversity	Diverse Online Ensembles Detector (DOED)
Feature selection, Hafding's inequality	Ensemble decision trees for conceptual drift (EDTC)
Ensemble learning - Online learning - Class imbalance learning -OS-ELM - Sampling technique	Extreme Sequential Online Learning Machines (ESOS-ELM)
Adaptive Window - Online and Incremental Learning - Weighted Extreme Learning Machine	Metacognitive Online Sequential Extreme Learning Machine (MOS-ELM)
Exponential Weighted Moving Average Chart - Ensemble Learning - Error Rate Monitoring	Ensemble classifiers with drift detection (ECDD)
Online classification ensemble method-Using the timing control chart	Window Dynamic Weighted Majority (DWM-WIN))
Ensemble Learning - Adaptive Random Forests Algorithm - ADWIN - PHT	Adaptive Random Forest (ARF)
Heterogeneous Dynamic Weighted Majority - Heterogeneous Ensemble Learning - Dynamic Weighting - Learnerseed	Heterogeneous Dynamic Weighted Majority (HDWM)
Algorithm EDDM -The concept of knowledge transfer - Hidden Markov Model	Recursive Adaptive Classifier (RACE) ensemble
PH test -Multi - class problemsBlock-based hybrid ensemble	Online drift detector for k- class problem (ODDK)
Ensemble learning-Combining multiple statistical drift detection methods	Statistical Drift Detection Ensemble (sdde)
Online hypothesis testing - Online ensemble learning - Statistics independent of specific distribution	Ensemble drift detection (GDDM)
Semi-supervised learning - Ensemble learning - Kolmogorov-Smirnov test - Isometric prediction	Isometric prediction for semi-supervised classification on data streams (CPSSDS)
Ensemble Learning - Fast Deep Stacked Network (FDSN) - From Single-layer feed-forward neural networks (SLFN)	Fast and deep sequential online stack network (OSFDSN)
Transfer learning -Dynamic Ensemble Classifier - Weighted majority voting	Selected ensemble using transfer learning (SETL)
Three-state decision framework - Ensemble learning - A divide and conquer strategy - Density clustering	In-sample weighted ensemble learning based on tripartite decision (IWE-TWD)

References

[1] Gama, J. (2012). A survey on learning from data streams: current and future trends. Progress in Artificial Intelligence, 1, 45-55.

[2] Bahri, M. (2020). Improving IoT data stream analytics using summarization techniques (Doctoral dissertation, Institut Polytechnique de Paris)

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

[4] Polikar, R. (2012). Ensemble learning. Ensemble machine learning: Methods and applications, 1-34.

[5] Agrahari, S., & Singh, A. K. (2022). Concept drift detection in data stream mining: A literature review. Journal of King Saud University-Computer and Information Sciences, 34(10), 9523-9540.

[6] Liu, H., Gegov, A., Cocea, M., Liu, H., Gegov, A., &Cocea, M. (2016). Ensemble learning approaches. Rule-based systems for big data: a machine learning approach, 63-73.

[7] Ravichandran, T., Gavahi, K., Ponnambalam, K., Burtea, V., & Mousavi, SJ (2021). Ensemble-based machine learning approach for improved leak detection in water mains. Journal of Hydroinformatics, 23(2), 307-323.

[8] Littlestone, N., & Warmuth, MK (1994). The weighted majority algorithm. Information and computation, 108(2), 212-261.

[9] Street, WN, & 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).

[10] Wang, H., Fan, W., Yu, PS, & Han, J. (2003, August). Mining concept-drifting data streams using ensemble classifiers. In Proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining (pp. 226-235).

[11] Kolter, JZ, & Maloof, MA (2007). Dynamic weighted majority: An ensemble method for drifting concepts. The Journal of Machine Learning Research, 8, 2755-2790.

[12] Elwell, R., &Polikar, R. (2009, June). Incremental learning in nonstationary environments with controlled forgetting. In 2009 international joint conference on neural networks (pp. 771-778). IEEE.

[13] Gama, J., Medas, P., Castillo, G., & Rodrigues, P. (2004). Learning with drift detection. In Advances in Artificial Intelligence–SBIA 2004: 17th Brazilian Symposium on Artificial Intelligence, Sao Luis, Maranhao, Brazil, September 29-October 1, 2004..

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

[15] Bifet, A., &Gavalda, R. (2009). Adaptive learning from evolving data streams. In Advances in Intelligent Data Analysis VIII: 8th International Symposium on Intelligent Data Analysis, IDA 2009, Lyon, France,

[16] Ditzler, G., &Polikar, R. (2011, April). Hellinger distance based drift detection for nonstationary environments. In 2011 IEEE symposium on computational intelligence in dynamic and uncertain environments (CIDUE) (pp. 41-48). IEEE

[17] Minku, LL, & Yao, X. (2011). DDD: A new ensemble approach for dealing with concept drift. IEEE transactions on knowledge and data engineering, 24(4), 619-633.

[18] Brzeziński, D., & Stefanowski, J. (2011, May). Accuracy updated ensemble for data streams with concept drift. In International conference on hybrid artificial intelligence systems (pp. 155-163). Berlin, Heidelberg: Springer Berlin Heidelberg.

[19] Ang, HH, Gopalkrishnan, V., Zliobaite, I., Pechenizkiy, M., & Hoi, SC (2012). Predictive handling of asynchronous concept drifts in distributed environments. IEEE Transactions on Knowledge and Data Engineering, 25(10), 2343-2355.

[20] Woźniak, M., Kasprzak, A., & Cal, P. (2013). Weighted aging classifier ensemble for the incremental drifted data streams. In Flexible Query Answering Systems: 10th International Conference, FQAS 2013, Granada, Spain, September 18-20, 2013. Proceedings 10 (pp. 579-588). Springer Berlin Heidelberg.

[21] Brzezinski, D., & Stefanowski, J. (2013). Reacting to different types of concept drift: The accuracy updated ensemble algorithm. IEEE transactions on

neural networks and learning systems, 25(1), 81-94
[22] Brzezinski, D., & Stefanowski, J. (2014). Combining block-based and online methods in learning ensembles from concept drifting data streams. Information Sciences, 265, 50-67.
[23] Liao, J. W., & Dai, B. R. (2014, May). An ensemble learning approach for concept drift. In 2014 International Conference on Information Science & Applications (ICISA) (pp. 1-4). IEEE.
[24] Maciel, BIF, Santos, SGTC, & Barros, RSM (2015, November). A lightweight concept drift detection ensemble. In 2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI) (pp. 1061-1068). IEEE
[25] Soares, SG, & Araújo, R. (2015). An on-line weighted ensemble of regressor models to handle concept drifts. Engineering Applications of Artificial Intelligence, 37, 392-406.
[26] Sidhu, P., & Bhatia, MPS (2015). An online ensembles approach for handling concept drift in data streams: diversified online ensembles detection. International Journal of Machine Learning and Cybernetics, 6(6), 883-909
[27] Li, P., Wu, X., Hu, X., & Wang, H. (2015). Learning concept-drifting data streams with random ensemble decision trees. Neurocomputing, 166, 68-83.
[28] Mirza, B., Lin, Z., & Liu, N. (2015). Ensemble of subset online sequential extreme learning machine for class imbalance and concept drift. Neurocomputing, 149, 316-329.
[29] Mejri, D., Limam, M., & Weihs, C. (2016). Monitoring a Dynamic Weighted Majority Method Based on Datasets with Concept Drift. In Analysis of Large and Complex. Data (pp. 241-250). Springer International Publishing.
[30] Dehghan, M., Beigy, H., &ZareMoodi, P. (2016). A novel concept drift detection method in data streams using ensemble classifiers. Intelligent Data Analysis, 20(6), 1329-1350. ,
[31] Haque, A., Khan, L., Baron, M., Thuraisingham, B., & Aggarwal, C. (2016, May). Efficient handling of concept drift and concept evolution over stream data. In 2016 IEEE 32nd International Conference on Data Engineering (ICDE) (pp. 481-492). IEEE. ,
[32] Gomes, HM, Barddal, JP, Enembreck, F., &Bifet, A. (2017). A survey on ensemble learning for data stream classification. ACM Computing Surveys (CSUR), 50(2), 1-36.
[33] Ren, S., Liao, B., Zhu, W., & Li, K. (2018). Knowledge-maximized ensemble algorithm for different types of concept drift. Information Sciences, 430, 261-281
[34] Sidhu, P., & Bhatia, MPS (2019). A two ensemble system to handle the concept of drifting data streams: recurring dynamic weighted majority. International journal of machine learning and cybernetics, 10, 563-578. ,
[35] Idrees, MM, Minku, LL, Stahl, F., &Badii, A. (2020). A heterogeneous online learning ensemble for non-stationary environments. Knowledge-based systems, 188, 104983
[36] da Silva, BLS, Inaba, FK, Salles, EOT, & Ciarelli, PM (2020). Fast deep stacked networks based on extreme learning machine applied to regression problems. Neural Networks, 131, 14-28
[37] Museba, T., Nelwamondo, F., Ouahada, K., & Akinola, A. (2021). Recurrent adaptive classifier ensemble for handling recurring concept drifts. Applied Computational Intelligence and Soft Computing, 2021(1), 5533777.
[38] Mahdi, O. A., Pardede, E., & Ali, N. (2021). A hybrid block-based ensemble framework for the multi-class problem to react to different types of drifts. Cluster Computing, 24(3), 2327-2340.
[39] Komorniczak, J., Zyblewski, P., &Ksieniewicz, P. (2022). Statistical drift detection ensemble for batch processing of data streams. Knowledge-Based Systems, 252, 109380.
[40] Xu, L., Ding, X., Peng, H., Zhao, D., & Li, X. (2023). Adtcd: An adaptive anomaly detection approach towards the concept of dr.
[41] Tanha, J., Samadi, N., Abdi, Y., & Razzaghi-Asl, N. (2022). CPSSDS: conformal prediction for semi-supervised classification on data streams. Information Sciences, 584, 212-234.
[42] da Silva, BLS, & Ciarelli, PM (2024). A fast online stacked regressor to handle concept drifts. Engineering Applications of Artificial Intelligence, 131, 107757.
[43] Cai, S., Zhao, Y., Hu, Y., Wu, J., Wu, J., Zhang, G., ... &Sosu, RNA (2024). CD-BTMSE: A Concept Drift detection model based on Bidirectional Temporal Convolutional Network and Multi-Stacking Ensemble learning. Knowledge-Based Systems, 294, 111681

[44] Arora, S., Rani, R., & Saxena, N. (2024). SETL: a transfer learning based dynamic ensemble classifier for concept drift detection in streaming data. Cluster Computing , 27 (3), 3417-3432.

[45] Deng, D., Shen, W., Deng, Z., Li, T., & Liu, A. (2025). An Ensemble Learning Model Based on Three-Way Decision for Concept Drift Adaptation. Tsinghua Science and Technology , 30 (5), 2029-2047.

Appendix:

Table 1. Timeline of ensemble drift detection methods

Algorithms with citation	Year	Key points	Benefits	Limitations
SEA	Stream Wakeim (2001)	Refine learners based on performance predictions- Processing Batch input data - Replacing the new category with one of the existing categories in the ensemble to adapt to the drift	Using majority voting – using fixed memory – adapting to speed with concept drift	Not suitable for dealing with the spread of semi-regulatory issues
AWE	Wang et al. (2003)	Adjust the weight of the models based on accuracy - use From estimating accuracy with time lag - selecting the best learners using a special version of the mean squared error - weighting each classifier using a special version of the mean squared error in the most recent chunk using cross validation	Improving the performance of classifiers in non-stationary environments with concept drift by weighting the base classifiers based on their accuracy in recent data	Weakness In iterative concepts - It has problems with missing or late-arriving class labels due to using prior accuracy for its pruning strategy - It requires a different pruning strategy to extend it to deal with semi-supervised data - Using cross-validation to calculate weights increases the AWE execution time.
ACE	Nishida and Yamauchi (2007)	Drift detection mechanism based onClassifiers accuracy reduction - includes an online classifier, a ensemble of parallel batch classifiers to increase prediction accuracy	Reuse of past knowledge - Ability to deal with repetitive drift . - Dynamics and flexibility - Continuously update the	Computational complexity - failure to consider sudden and gradual incremental drift

			model with new data	
DWM	Coulter and Maloof (2007)	Online learning - WMA -inspired weighting - Using four mechanisms: online learning of learners, weighting and weight adjustment based on performance, dynamic addition and removal of weak learners	Flexibility - dynamically adapts to conceptual changes in data	Failure to consider specific types of concept drift - need for adjustment Weight threshold foot meter
Learn++.NSE	Elwell and Pallikar (2009)	Incremental Learning - Learn++ Improvement - Dynamic weighted voting based on model accuracy – Training andcombining new classifiers with error- and age-based weighted voting	Suitable for non-static environments - Optimized memory usage - Adaptable to concept drift at variable rates	Delay in detecting new drift - complexity in implementation
AUE	Brzezinski and Stefanowski (2011)	Improving AWE by conditional updating learners instead of adjusting weights - Online learning - Gradual updating of models incrementally - Combining accuracy and diversity - Determining learner weights based on mean square error	Better performance than AWE - good balance between accuracy/versatility	Require constant processing time and memory - High computational cost - Complex implementation
Learn++.CDS	Ditzler (2011)	Learn++ - Incremental Learning- Combining the Learn++.NSE algorithm for learning from concept drift with the SMOTE technique for dealing with class imbalance	Tackling class imbalance - monitoring both the majority and minority classes	SMOTE computational cost - probability of generating noisy samples - complexity in implementation
Learn++.NIE	Ditzler (2011)	Learn++ - Incremental learning - Adaptive penalty to balance accuracy in minority and majority classes - Adjust sample weights to balance recall across classes	Suitable for unstable and unbalanced environments	Delay in detecting new drift - complexity in implementation

DDD	Monica and Yao (2011)	Online learning - Controlling learner diversity by combining low- and high-diversity ensembles - Adjusting learner weight based on accuracy at different times - Using the low-diversity ensemble to detect drift and the high-diversity ensemble afterwards .	- Higher accuracy than EDDM - Significant resistance to false alarms - Better accuracy in stable concepts than EDDM - Higher accuracy than DDDDWM	Computationally expensive - Complex diversity criteria - No use of long-term memory
PINE	Ang et al. (2012)	An ensemble approach for asynchronous concept classification in distributed networks with a modified version of ADWIN for accuracy-based flow drift detection	Low sensitivity to parameters - reduced communication cost - higher accuracy	Computational complexity - need to adjust parameters
WAE	Woznick et al. (2013)	Time-based and age-based weighting inspired by AWE ; adding classifications based on accuracy and diversity	Adaptability, use of incremental learning, high efficiency	Computational complexity
AUE2	Brzezinski and Stefanowski (2013)	AUE Development - Combining block-accuracy-based weighting mechanisms with the incremental nature of Hafding trees	Stronger than AUE - Less memory usage	Higher complexity - Slower adaptation in some cases
OAUE	Brzezinski and Stefanowski (2014)	-Online boot diagnostic - Weighting classifiers based on error with fixed memory; using windowing technique and new incremental error-based weighting function	Detection of sudden, gradual drifts	Does not detect all drifts
DOED	Sidhu and Bhatia (2015)	Identifying drift in diverse online ensembles by comparing accuracy and statistical testing	Better accuracy in detecting sudden and gradual drift- Computational efficiency	Failure to detect repetitive and incremental drift
OWE	Suarez and Araujo (2015)	Online weighted ensemble with gradual learning of regression models; moving window for	Retaining past information	Computational complexity - choosing the window size

		new samples; error determination with boosting strategy ; weight assignment with discount factor to handle repetitive changes .		
ESOS-ELM	Mirza and colleagues (2015)	Learning class imbalance-A ensemble of OS-ELMs - Creating class balance with sampling techniques on training data	Drift-free online learning for unbalanced data using prior knowledge.	Sensitive to hyperparameter s - may be over-fitting
DWM-WIN	Mujeri et al. (2016)	Online classification ensemble method with dynamic weighted majority; using control chart to monitor error rate and detect drift	Performance Monitoring - Online Learning	Computational integration - Need to adjustcontrol chart parameters
ARF	Gomez et al. (2017)	Using Parallelization of Adaptive Random Forests Algorithm – Using ADWIN and PHT for Drift Detection	High accuracy - Dealing with all kinds of conceptual drift without complex optimization	High memory usage - complex to implement
Mixed forest	Rad and Haeri, 2019	Identifying weak learners in classification and regression	Low latency and fast startup	More efficient computing resources are needed .
RDWM	Sidhu and Bhatia (2019)	DWM- based recurrent weighted majority with two ensembles of learners: an initial ensemble for current concepts and a secondary ensemble of the most accurate learners.	Managing recurring conceptual drifts -Very high generalization accuracy	Poor performance on non-repetitive conceptual drifts in prediction
HDWM	Idris et al. (2020)	Heterogeneous dynamic weighted majority online learning; using diverse base learners and seed learners to maintain diversity; dynamically adjusting weights based on performance .	Better performance in non-static environments than WMA - Improved performance in detecting recurring conceptual drifts Compared to DWM	Non-repetitive conceptual drifts not investigated - need for fine-tuning of parameters
RACE	Mousbeh et al. (2021)	Recursive adaptive classifier ensemble- Maintaining an archive of diverse learners and using	Managing Recurring Concepts - Combining Knowledge	Other concepts of concept drift are not considered.

		EDDM to identify recurring trends	Transfer and Drift Detection - Improving Algorithm Prediction Accuracy for Non-Stationary Time Series Data	
ODDK	Mehdi et al. (2021)	Ensemble based on combined block-Handling multi- class problems with concept drift-PH test to detect conceptual drift	Integrating the main advantages of an online drift detector for a k-class problem and block-based weighting.	Does not recognize all drifts
CPSSDS	Tanha et al. (2022)	Incremental base learner in a self-learning framework; Unlabeled sample selection with isomorphism prediction; Concept drift detection with Kolmogorov - Smirnov test .	Better performance in limited-function labeling conditions comparable to advanced semi-supervised algorithms	Dependence on the selection of unlabeled samples - computational complexity
GDDM	Yu et al .(2023)	Drift detection method based on error rate by developing online hypothesis testing based on a new statistic, independent of data distribution and dimensions; determining the ensemble drift threshold without considering the correlation of flows; a dynamic threshold that adapts to environmental changes instead of a fixed value .	No dependence on data distribution; online operation without the need for full data storage	Computational complexity - need to adjust parameters
OSFDSN	Da Silva and Ciarelli (2024)	FDSN Single-layer Stacked Feedforward Neural Networks with the latest SLFN output ; OSFDSN Online version with dynamic module weighting based on new data	High accuracy and faster update than other methods; equivalent statistical performance in RMSE ; FDSN feature combination in the face of concept drift	Has some of the problems of FDSN - Structure compactness - Maintaining multiple models in a stack structure increases memory consumption

CD-BTMSE	Kai et al. (2024)	Concept drift detection using Bidirectional Temporal Convolutional Network (BiTCN) and Multi-Stacking ensemble learning to improve accuracy	Addressing the weakness of generalization of ensemble learning methods by replacing ReLU with LeakyReLU and replacing the negative log-likelihood function with focal loss in the BiTCN model	Requiresre latively high computati onal resources - Dependen cyTo the quality of educationa l data
SETL	Aurora et al. (2024)	Transfer learning - Dynamic Ensemble Classifier - Weighted majority voting	Overcoming negative transfer and overfitting	Computational complexity and the need for fine-tuning parameters - dependence on source data quality in transfer learning .
IWE-TWD	Deng et al. (2025)	Three-state decision framework with ensemble learning, dividing the decision space into positive, negative and boundary regions; dynamic adaptation to drift through density clustering; selection of diverse learners and weighting of samples based on the probability of distribution change	Effectively managing concept drift uncertainty; solving problems of adapting local drifts and different decision boundaries; avoiding using inappropriate general measures of diversity .	High computati onal cost - Dependen ce on the type of conceptch ange