Journal of King Saud University – Computer and Information Sciences 34 (2022) 9523–9540

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

Concept Drift Detection in Data Stream Mining : A literature review

Supriya Agrahari [a,*], Anil Kumar Singh [a]

[a] Motilal Nehru National Institute of Technology Allahabad, Prayagraj, India

ARTICLE INFO

Article history:
Received 12 August 2021
Accepted 6 November 2021
Available online 3 December 2021

Keywords:
Concept drift
Concept evolution
Adaptation mechanism
Data stream mining

ABSTRACT

In recent years, the availability of time series streaming information has been growing enormously. Learning from real-time data has been receiving increasingly more attention since the last decade. Online learning encounters the change in the distribution of data while extracting considerable information from data streams. Hidden data contexts, which are not known to the learning algorithms, are known as concept drift. Classifier classifies incoming instances using past training instances of the data stream. The accuracy of the classifier deteriorates because of the concept drift. The traditional classifiers are not expected to learn the patterns in a non-stationary distribution of data. For any real-time use, the classifier needs to detect the concept drift and adapts over time. In the real-time scenario, we have to deal with semi-supervised and unsupervised data, which provide no or fewer labeled data. The motivation behind this paper is to introduce a survey identified with a broad categorization of concept drift detectors with their key points, limitations, and advantages. Eventually, the article suggests research trends, research challenges, and future work. The adaptive mechanisms are also incorporated in this survey.

Contents

* Corresponding author.
E-mail address: supriyaagrahari@mnnit.ac.in (S. Agrahari).

https://doi.org/10.1016/j.jksuci.2021.11.006
1319-1578/

1. Introduction

In the digital world, the data generate from various applications and grow at a rapid rate. The world is changing with the growth of technologies. Any sequence of data with a timestamp is known as a data stream. The data stream contains a huge volume of data with varying velocities. Digital data generated from different sources have the following characteristics:

- Data streams are unbounded in size.
- Data arrives at a high rate and varying velocity.
- Data may evolve over time.
- Data processing has constraints of memory.

The data stream is not finite, i.e., we do not know how much data in the sequence are left. Nowadays, with various advancements in technology, different applications generate vast amounts of data at a very high speed. Over time, the generated data must be analyzed near real-time as one hardly has a chance to look at the data before it passes. In a data stream, the problem of memory constraint must be resolved because real-time data requires storage for its processing. The data is stored on a hard disk in traditional systems, whereas the data streams mostly need to be processed in the main memory. The data stream exhibits non-stationary distribution because the data may change over time. Table 1 shows the various aspects of traditional data mining and data stream mining. The order of observation makes over the data in a stream that has significant learning. So, the change over data distribution needs to be detected. This detection is one of the major challenges in data stream mining. In the data stream, the data points are generated sequentially and independently under some probability distribution Kifer et al., 2004. Data streams have several requirements as follows:

- Process one example at most once.
- There is a memory restriction for the learning of an example.
- Required to process in a limited amount of time.
- Ready to predict the class labels whenever required.

The data stream learning is performed in a supervised, semi-supervised, and unsupervised fashion. Supervised learning needs to anticipate the target labels for classifying the data stream, and other techniques require less or no labeled data. Fig. 1 shows the abstract view of model building in the data stream. It has three modules. The first module is the input module, which takes training data instances or examples as input, and the algorithms provide this input. The incoming data instances are stored in memory and processed subsequently because the data streams instances are arriving continuously. Still, there is a limitation of space and time. The second module is the learning module, which learns from the current data pattern and provides the basis of model building in the third step. Finally, in the third module, the initial model is built. The model takes new incoming data instances and predicts their labels. The above whole process will repeat till the data become available. The classification of data stream manifests several challenges Gama, 2010:

- Data streams arrive from single or multiple sources with varying velocities.
- Arrival of the data stream has no order.
- We can not store all the information of data, but it can store summary.

The factors like speed and size of data are considered as one of the problems in non-stationary distribution. Accordingly, the researchers have solved these problems. Still, there are more complex areas, which deal with the dynamic distribution of data.

Table 1
Traditional data mining Vs Data stream mining

	Number of passes	Time requirement	Memory requirement	Number of concepts	Result obtained
Traditional data mining	Multiple	Unlimited	Unlimited	One	Accurate
Data stream mining	Single	Limited (Real-time)	Limited	More-than-one	Approximate

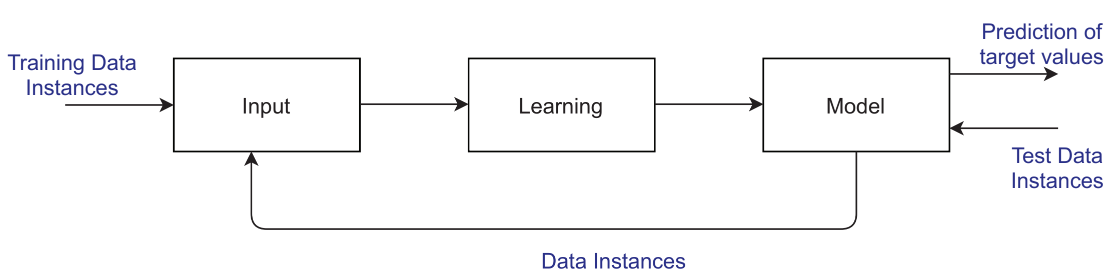

Fig. 1. Abstract view of model building in data stream mining.

The identification of concept change is relevant in the context of the data stream that offers various challenges. An example of concept drift is the change in user choices following an online movie stream. The distribution of movies is often broadcasted that might remain the same. Still, the conditional distribution of exciting movies for the user might change. Due to the change in concept over real-time data, adaptive learning is necessary. The adaptive learning reflects the predictive models online while their operation responds to the concept drifts. Real-time learning generally discards the old concepts and adapts the new concepts. So, an adaptive learning algorithm requires to handle concept drift. Forgetting mechanism and window mechanism are the most important prequential to endure and evolve through concept drift Khamassi et al., 2018. Here, the main objective is to build a model to identify the changes in data instances and adapt the data distribution changes.

The shift or change in data distribution could be sudden, gradual, and recurring. Sudden drift occurs at a particular time instance where the change in the distribution of data is done suddenly with no time overlap between them. If there is an overlap between the change in concept, it is considered as gradual drift. A recurring class is one of the special conditions of concept drift when the same concept is seen after a long time. Sudden drift is comparatively easier to identify than gradual and recurring drift.

Concept drift can be seen in different domains where the predictions are ordered by time (see Fig. 2). For example, the weather forecast has three attributes: temperature, humidity, and pressure. The season may be one concept in the weather data stream that impacts temperature (i.e., it is not explicitly specified in temperature data), but it may influence the weather data. Here, the hidden information can cause concept drift. Another example is the customer purchasing behavior over time that may influence the economy's strength of a country, where the economy's strength is not explicitly specified in the purchasing data. Here, the economy's strength is a hidden context in customer purchasing behavior data. In this way, the data collected within a specific period shows the change in the relationship of customer purchasing behavior with the economy's strength. Thus, the concept drift refers to the change in the relationship concerning time. The main challenges of concept drift in data streams are:

- Due to concept drifts, the performance of the classifier deteriorates with the change in time. So, it becomes obsolete.
- Tracking concept change.
- Adaptation of classification model as per the current data instances to react in the non-stationary distribution of data.
- Incorporating window and forgetting mechanism with change in concept.

Different researchers have built different algorithms to detect the change in the data stream by calculating the distance between distributions, prediction errors of incoming data instances, and changes in the classifier's accuracy. The degradation in accuracy shows a change in the distribution of data. Sometimes due to sensor failure, noise, and outlier, the accuracy of the classifier is also degraded. So, the distinction among them is necessary to detect actual concept drift in the incoming data stream. The classifier's accuracy reduces because it builds as per the previous data with different distribution than current data. So, the distribution is independent before and after the change point. The online algorithm finds the most recent change in the stream and accordingly accommodates their model.

The real-time learning algorithm must adapt as per the incoming instances of the data streams, known as adaptive learning. It is an extension of an incremental learning algorithm. Incremental learning is one of the methods of machine learning. It continuously takes input data and extracts the knowledge from the data. It performs learning and adapts the model accordingly. Incremental learning is a dynamic technique for supervised and unsupervised learning. It is applied where the memory size is overflowed because of the limited system memory, and the gradual data stream is coming over time. There are many incremental learning algorithms such as incremental Support Vector Machine (SVM) Cauwenberghs et al., 2001, Decision Tree (DT) Rasoul Safavian and Landgrebe, 1991, Artificial Neural Network (ANN) Basheer and Hajmeer, 2000, Decision Rule (DR) Hart, 1968, etc.

The drift detector detects the change position; thereby, it replaces the base learner to improve the overall accuracy of the learning model. Generally, the drift detectors are configured with fully labeled data. They usually use two different base classifiers, namely Naive Bayes and Hoeffding Tree. These classifiers are also defined in MOA framework Barros et al., 2018. Concept drift detectors are used to find the drift near its occurrence point. Several synthetic and real-time data are used to verify the working of different detectors.

The contributions of the paper are:

- It broadly categorizes the concept drift detectors and identifies new categories such as similarity and dissimilarity based methods, significance analysis based methods, decision boundary based methods, etc.
- It chronologically present the detectors with their keypoints, limitations and advantages.
- It suggests research trends, and research challenges in concept drift detection in the streaming environment.
- It describes the potential research directions for future concept drift detection prospective.

The paper is organized as follows: Section 2 discusses Concept Drift and their types. Several existing drift detection algorithms and their various categories are illustrated in Section 3. The detailed description of Concept Drift Adaptation discusses in

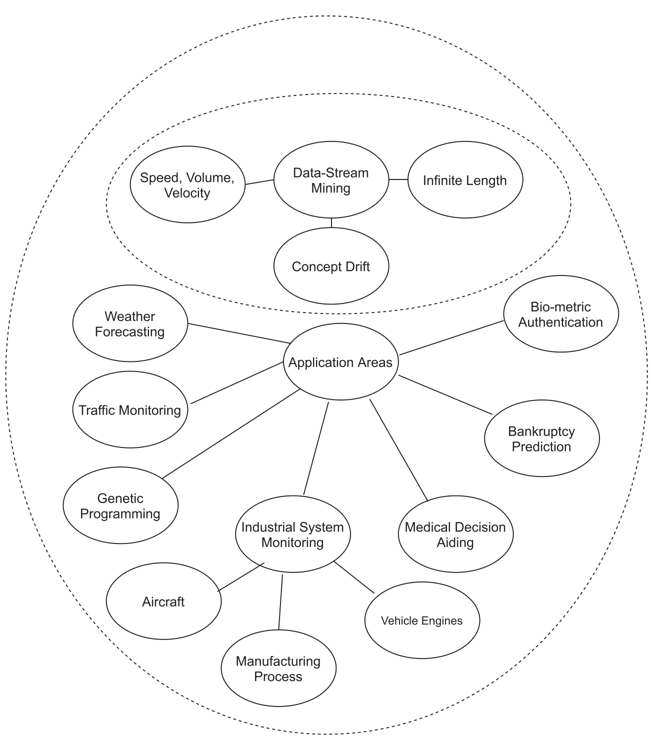

Fig. 2. Data stream mining and their application areas.

Section 4. The Research Trends are present in Section 5. Section 6 describes the Research Challenges. The Conclusion is in Section 7, and Future Work is discussed in Section 8.

2. Concept drift

The data stream is continuous time-series data of infinite length. The online methods are required to learn these data streams. Streaming data generally encounters the limitation of space and time Compared with batch processing (having multiple scans), the real-time non-stationary data stream distribution has only one scan.

Concept drift occurs when the underlying data distribution changes over time Ryan Hoens et al., 2012. Suppose the data stream at timestamp t consists of n labeled data instances, i.e., $\{(X_{i_t}, c_{i_t}), \ldots, (X_{n_t}, c_{n_t})\}$. Each instance has m feature vectors, i.e., $\{a_i, \ldots, a_m\}$ and the corresponding attribute values at t timestamp is described as $\{(a_{i_t,j_t}), \ldots, (a_{i_t,m_t})\}$. The class label is denoted by C and it belongs to the set of labels, i.e., $\{c_{i_t}, \ldots, c_{n_t}\}$.

According to Bayes' theorem, the following measures are taken into consideration while finding the concept drift Gama et al., 2014:

- $p(c)$ may change, it means that the class prior probability may change.
- $p(X_{t+1}|c)$, i.e., the distribution of classes may change.
- $p(c|X_{t+1})$, i.e., the posterior probability distribution of class membership may change.

The first measure reduces the classifier's efficiency that is leading to a class imbalance problem. This problem occurs when the total number of one class of data is far less than the second class (in a binary class problem). The second measure is virtual drift, and the real drift happens in the third measure. In virtual drift, the decision boundary does not change with a change in the data distribution. In real drift, the decision boundary changes concerning the change in data distribution with time.

Generally, the concepts formalize in terms of label, intention, and extension. The intention of the concept defines the properties implied by it. In contrast, the extension refers to the set of things that it extends, and the labeling is used to provide the concept's meaning in natural language description. Finally, the concepts (label(C), intention(C), extension(C)) are core properties that can be uniquely identified and do not change over time Demšar and Bosnić, 2018; Wang et al., 2010; Geoffrey et al., 2016. The concepts C1 and C2 are identical, if their intentions are equivalent, i.e., $intention(C_1)$= $intention(C_2)$. Suppose a concept has the same meaning at different times. It shows no concept drift. For example, the grades of a student lie between some percentage range. Previously, the grade 'A' meant that the percentage lies between 70-80%, but now it defines between 80-90%. So, there is a change in the meaning of grade 'A' with time, representing a change in concept.

2.1. Types of drifts

Data streams are continuous, and the distribution of real-time data is non-stationary. The data distribution may differ over time. These changes in the data can be considered, namely real concept drift and virtual concept drift, as two types of drifts. Some researchers define the class prior as one of the drift. The drift can be abrupt (or sudden), gradual, incremental, recurring, and blip in terms of speed of change in concept. However, the blip represents quick and sudden change (or rare event) in a concept, and it refers to an outlier in a stationary distribution. So, generally, it is not considered as drift. Fig. 3 and Fig. 4 describe different types

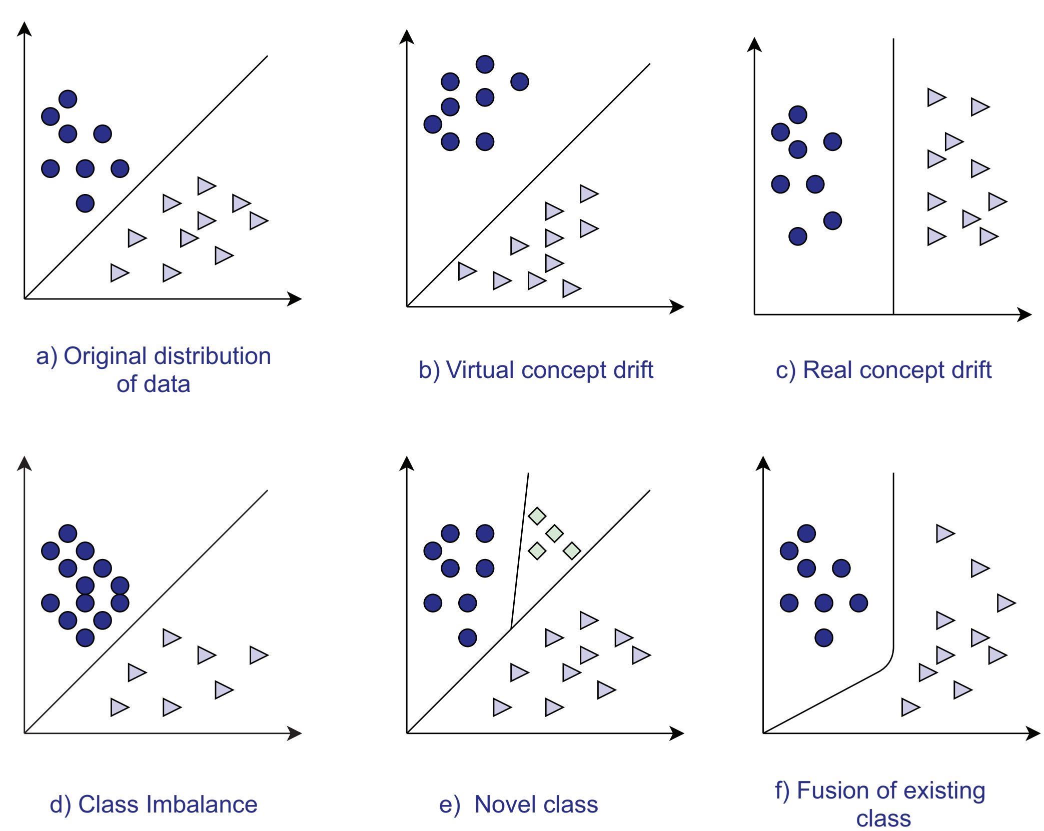

Fig. 3. Representation of different types of concept drifts.

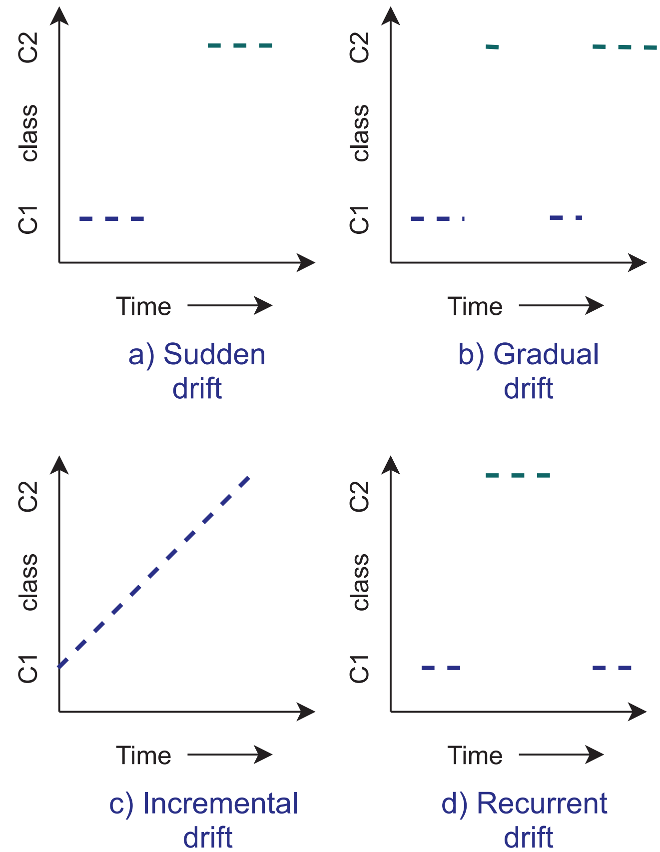

Fig. 4. Types of concept drifts in terms of speed.

of concept drifts and the types of drifts in terms of speed, respectively.

2.1.1. Real concept drift

There is a change in posterior probability $P(c|X)$ in real concept drift (see Fig. 3 (c)). It represents a change in the target concept with the same value of attributes as before. Due to this, the decision boundaries are affected, and thereby, it decreases the model performance.

2.1.2. Virtual concept drift

Virtual concept drift happens due to the change in class conditional probability $P(X|c)$ without affecting $P(c|X)$ (posterior probability) (see Fig. 3 (b)). The decision boundary remains the same. Still, it affects the data distribution within the same class. Virtual drift can cause a class imbalance problem, where the total number of data in class c_i is much lesser than another class c_j in the case of two-class (c_i, c_j) classification problem. For example, in fraud detection, the fraudulent transaction can be 1% of data from another side (non-fraudulent transaction) of class.

2.1.3. Class prior

It refers a change in $P(y)$ (prior probability) Khamassi et al., 2018. This kind of drift leads to the different phenomenon: Class imbalance (see Fig. 3 (d)), New class emergence (Novel class detection) (see Fig. 3 (e)) and Existing class fusion (Merge data points in the existing class) (see Fig. 3 (f)). Class priors is not an independent drift. The class imbalance causes virtual drift where there is no change in decision boundary.

2.1.4. Abrupt drift

Here, the new concept of the incoming data stream suddenly replaces the old concept. So, the point of time where the old concept change to the new concept suddenly is known as abrupt drift (see Fig. 4 (a)). It realizes when the classifier's accuracy is degraded suddenly. After that, the model learns quickly from the new concept's characteristics, and accordingly, the changes are adopted.

2.1.5. Gradual drift

In gradual drift, the concept change duration is relatively large as compared to sudden drift (see Fig. 4 (b)). There are two variations in this type of drift: slow gradual drift and normal gradual drift. For example, the gradual drift is seen when the market phenomena change due to inflation or recession. This kind of drift has an overlapping concept, and after some period of time, the new concept becomes stable.

2.1.6. Recurrent drift

In this kind of drift, the concept reappears after a long period of time, i.e., a recurring change in concept happens in the stream (see Fig. 4 (d)). It has cyclic and acyclic behavior. The cyclic phenomenon applies in a situation where seasonal variations occur. For example, the selling of cold items is increased due to the summer season. The acyclic phenomenon is seen where the price of electricity is increased due to the rise in petrol price and in a normal situation it returns to the previous price.

Several types of drifts happen in different scenarios, and it depends on various applications. Many algorithms (drift detectors) are already built to find the drift near its closest point of occurrence. Some detectors are working well for sudden or gradual or recurring drift or combination of them. Still, there are some limitations to the existing detectors. So, these areas need to explore.

The learning algorithm works according to the nature of the incoming data set. Generally, the analysis of data instances is done by batch processing and non-stationary data processing. The batch processing occurs in stationary data distribution, whereas the adaptive learning method is used in the non-stationary data distribution. Batch learning algorithms are trained with the non-stream learner. Learner ignores all new data and focuses on the previously learned concept. In non-stationary data distribution, the learner immediately adapts to the latest instance and discards the old instance Ryan Hoens et al., 2012. The learning algorithm is incremental when it satisfies the following criteria:

- The sequence of hypotheses is produced from the data that describes all the data, which has been seen so far.
- It produces a sequence of hypotheses like the current hypothesis that describes the current data.

The learning algorithms require a classifier that updates incrementally based on newly available data, and it has to maintain the classifier's performance simultaneously. The old learning algorithms are dependent on the existing data, whereas the incremental algorithms update based on the current data Ryan Hoens et al., 2012.

Concept drift and Concept shift are two different terms related to data stream mining. The gradual change in the concept differentiates both. In the case of concept shift, a change between two concepts are more abrupt than concept drift. Distribution change is generally observed in the data stream. It occurs due to the sampling change or shift or virtual concept drift in the data. Distribution of data means that the change is happening underneath. The other possibility of distribution change is to modify the current model where the model's error rate may no longer be consistent with the new data distribution even if the concept remains the same Ahmad et al., 2017; González et al., 2019.

There exist some fundamental limitations in drift detection. Drift detectors do not detect the drift in the first position. That's why it deals with false alarms and compromising the classifier's accuracy. The behavior of a data stream learning algorithm needs to observe the above limitation. There are dimensions of interest for drift detection algorithms: 1) Memory requirement, 2) Less time to learn from training examples for label prediction of new examples using an existing classifier, 3) The prediction errors of labels Ditzler and Polikar, 2012.

Sebastiao and Gama (2009) categorize the drift detection methods into four sub-categories: 1) Sequential Analysis based Methods, 2) Adaptive Windowing, 3) Fixed Cumulative Windowing Schemes, 4) Statistical Methods. The sequential analysis based methods are sequentially predicted the drift based on evaluation and alarm the drift. Cumulative Sum (CUSUM) and Geometric Moving Average (GMA) are sequential analysis based methods. Windows-based methods are either use a fixed window or an adaptive (dynamic) window. It is usually summarizing the old information and the most recent (or current) information. The difference assessment between both information helps in detecting the drift. The examples of window-based methods are ADWIN, $HDDM_{A-test}$, and $HDDM_{W-test}$. The statistical methods are based on statistical parameters like mean, standard deviation, and variance to predict the drift. DDM, EDDM, EWMA, etc., are statistical based methods.

Adriana Sayuri Iwashita (2019) introduces an Optimum Path Forest (OPF) classifier. It can handle concept drift in three ways: a) Window-based approaches (fixed and dynamic window), b) Weight-based approaches (older instances are discarded based on weight), c) Classifiers' ensemble. This classifier is based on a graph partition task. It defines the predefined adjacency relation of the graph in which every node is encoded accordingly.

Elwell and Polikar (2011) develops Learn++ algorithm for non-stationary data distribution. It is an ensemble based approach for classification. It learns from the adjacent data batches. It can find the constant variable rate of drift as well as cyclic or recurrent drift. It uses weighted majority voting and simultaneously adjusts the accuracy as per the current and previous environment. This method gives an improved result with no pruning for a slow drift situation.

Masud et al. (2011) focuses on the recurring concept in novel class detection. Much of the computation and memory-intensive work is needed if a class is detected as novel but reoccurring. We know that the labeling of instances is also a costly process. This novel class detection algorithm recalls information about the previous classes. It addresses various data stream issues such as increased false alarm rates, unnecessary use of human effort, and additional computational effort.

Jie et al. (2020) deal with real-time data and pre-exist concept drift. It focuses on data-driven decision making which concerns a decision-based problem in a real-time environment. It deals with concept drift in terms of when, what, and how in a non-stationary environment. Here, when means the drift occurrence time; what defines the different types of drift present in the data stream; when and how the decision model adaptation is examined for concept drift situation.

Most of the research work is focused on some limitations of the data stream such as unbounded length, change in concept, the evolution of the new concept, and recurring concept Masud et al., 2011; Faria et al., 2016; Gama et al., 2004; Nishida, 2008; Wang et al., 2011. The concept drift detectors are used in various applications like detection of theft in the energy distribution system, churn prediction for mobile companies, fraud detection, etc. Many algorithms like CVFDT, SEA, AWE, VFDR, boosting methods, bagging with adaptive size holding tree, iOVFDT, WAE, AddExp, VFDTc, DWM, EAE, etc., predict and adapt the concept drift in the data stream. Still, these algorithms are not performed accurately with noisy data.

Due to continuously generated real-time data, the analysis of concept drift is very important. Learning from these real-time data is done by machine learning algorithm Ramírez-Gallego et al., 2017. Traditionally, the data stream instances organize in a static database, and use in batch processing. Nowadays, we deal with the non-stationary distribution of data. Also, the concepts may change with time. So, these aspects need to be focused on in the streaming environment.

3. Detection of concept drift

Traditional data mining approaches deal with a single concept where training and testing data has the same distribution, whereas data stream mining has multiple concepts with different data distributions Nguyen et al., 2015; Silva et al., 2013. Concept drift is a phenomenon in which the prior probabilities of the classes $p(c)$ discover the patterns and the class conditional density functions $p(X|c)$, where X is attribute vector, c is target variable, and $i = 1, \ldots, n$. The change in the above conditions is responsible for the change in underlying data. Fig. 5 illustrates the general block diagram of the concept drift detection method. It is divided into three phases. In the initial phase, initial data stream instances are used to build the learning model which predicts the target values. In the next phase, concept drift is identified in the data samples. If there is no drift in the data samples, further prediction of current data instances is performed. When drift is detected, the interpretation of concept drift is occurred, and successively concept drift adaptation and the forgetting mechanism are performed in the last phase. Here, in the adaptation process, it updates the learning model using current data instances. In this way, concept drift detection is performed in the streaming environment.

Several drift detection algorithms analyze the accuracy of classifier Barros et al., 2018. Since the classifier's accuracy degrades over time due to changes in the data distribution. Table 2 defines the algorithmic steps to deal with drift. Still, it needs to monitor. There are many ways to monitor concept drift, as given below:

- Concept drift detection is done by checking the probability distribution of data since it may change over time due to distribution, noise, outlier, sensors slowdown, etc.
- One can analyze whether the concept drift has happened. It can be done by monitoring and tracking the similarities or dissimilarities between different sample characteristics or attributes.
- Concept drift leads to degrading the accuracy of the classifier. So, it can be one of the measures while detecting the concept drift in a given data stream.
- Recall, Precision, F-measure, ROC, and AUC are some of the accuracy measures of a classification model. Generally, these measures are used by many authors.
- Timestamp can be used with either a single sample or block of samples. Timestamp can be one of the given input attributes to know the occurrence of concept drift.
- The features of classification models can be changed due to concept drifts Li et al., 2016.

3.1. Concept drift detectors

The concept is not stable because it changes over time. These changes make the model inconsistent. Thus, it is necessary to update a model regularly. The change in distribution concerning time may increase the errors in a learning model. So, the detection mechanism traces the errors online. In this article, the concept drifts detection algorithms are divided into several categories:

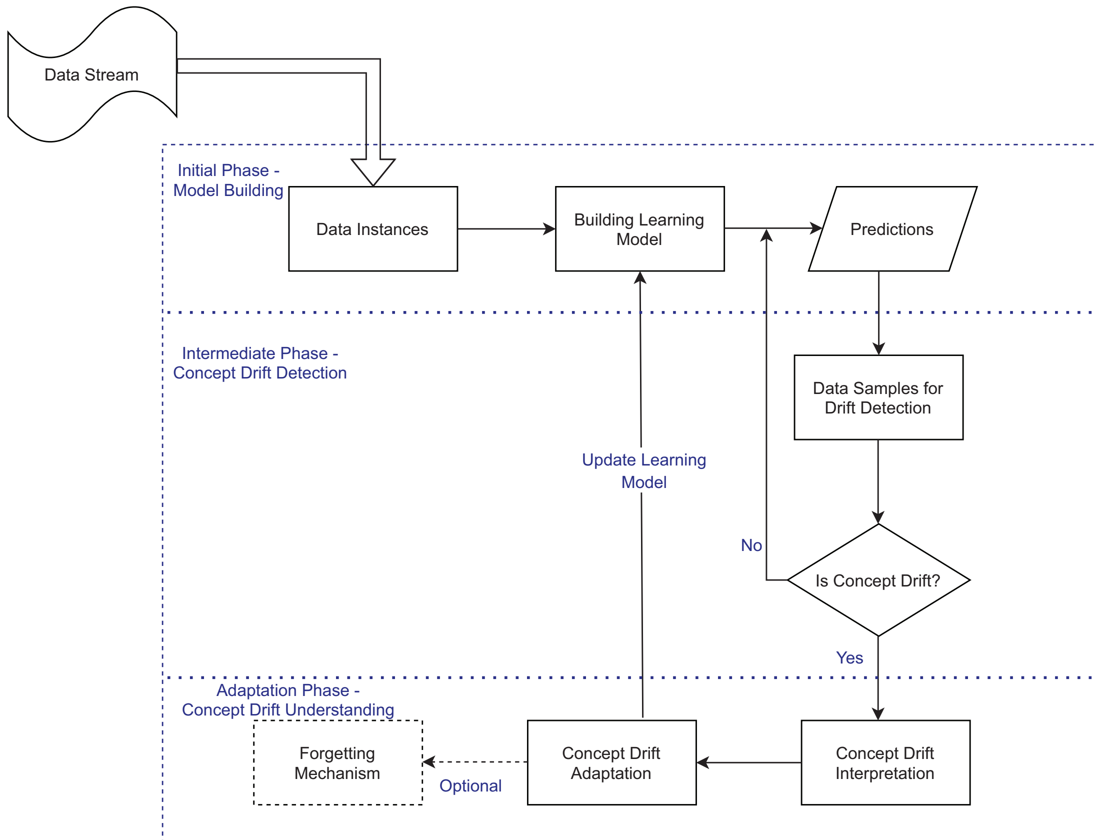

Fig. 5. General block diagram of concept drift detection.

Table 2
Algorithmic steps to handle concept drift

Steps No.	Name of Steps	Methodology
Step1:	Data processing	- Statistical based methods
		-Similarity and dissimilarity based methods
	methods	-Significance analysis based methods
		-Data distribution based methods
		-Decision boundary based methods
		-Model dependent based methods
		- Window based methods: Fixed and Adaptive windowing methods
		-Sequential analysis based methods
Step2:	Data learning process	- Single learner
		- Ensemble learner
Step3:	Concept drift monitoring methods	- Supervised methods
		- Unsupervised methods
		- Semi-supervised methods
Step4:	Concept drift handling	- Informed methods
		- Blind methods
Step5:	Performance	- Recall, Precision, F-measure, AUC, ROC
	measures	- Time and space complexity
		- Change in Accuracy
		- Parameter settings
		- Reliability

Similarity and dissimilarity based methods, Sequential analysis based methods, Window based methods, Statistical based methods, Significance analysis based methods, Data distribution based methods, Decision Boundary based methods, and Model dependent based methods.

3.1.1. *Similarity and dissimilarity based methods*

These approaches are based on measuring the similarity and dissimilarity among the distribution of data samples with respect to time. There are several drift detection algorithms built based on this approach that is discussed in this subsection.

The most popular approach is Drift Detection Method DDM Gama et al., 2004 and it is based on binomial distribution. Here, the data instance is available in the form of (X_i, c_i), where X is input attribute vector, c is target values, and 'i' is i^{th} sequence of data instance that is being sampled. The actual prediction of learning model is c'_i. The prediction of model is true when $c_i = c'_i$ and false when $c_i \neq c'_i$. The error rate is the probability of observe false p_i with standard deviation $s_i = \sqrt{(p_i(1-p_i))/i}$. If the probability distribution is stable, the confidence interval is taken as $1-\alpha/2$ and for p with $n > 30$ examples is approximately $p_i \pm \alpha * s_i$. The α depends on the confidence interval. The method defines two levels of concept drift detection. In case of warning level, $p_i + s_i \geqslant p_{min} + (2 * s_{min})$, and the confidence interval is set to 95%. Beyond the warning level (with the confidence interval is set to 99%), when the false detection rate is reached and $p_i + s_i \geqslant p_{min} + (3 * s_{min})$, it shows a drift. DDM is a learning algorithm independent method for drift detection. EDDM is similar to DDM. Here, the warning alarm mechanism is used as in DDM. Instead of using the classifier's error rate, the method uses a distance error rate. The main difference between both of them is that DDM gives good accuracy when there is a sudden and gradual change in concept. In comparison, EDDM performs well for a slow, gradual change. In EDDM, the distance between classification errors is used to detect the drift. It calculates the average distance between error and standard deviation. The average distance between two errors and the standard deviation is denoted as p'_i and s'_i, respectively. The values of p'_i and s'_i are stored when the value of $p'_i + (2 * s'_i)$ at its maximum value and then it obtains p'_{imax} and s'_{imax}. So, $p'_{imax} + (2 * s'_{imax})$ shows the corresponding point where the distribution of distance between errors is maximum.

Here, the warning level is denoted as $(p\prime_i + (2 * s\prime_i)) / (p\prime_{imax} + (2 * s\prime_{imax})) < \alpha$. It means that the concept drift may have happened. Drift level is obtained when $(p\prime_i + (2 * s\prime_i))/ (p\prime_{imax} + (2 * s\prime_{imax})) < \beta$. It shows that drift is confirmed. In case of concept drift, the learning model is reset, and for further processing, $p\prime_{imax}$ and $s\prime_{imax}$ are also reset. The author considers a drift when the 30 number of classification errors occur. Here, the value of alpha and beta are 0.95 and 0.90, respectively. EDDM does not increase the false positive rate and detects the drift faster compared to the DDM. False positive rate is stored in "look-up table" to monitor the drift in the stream. EDDM works better than DDM for slow gradual drift and is more robust to noise Baena-Garcıa et al., 2006. The disadvantage of this method is that it takes 30 errors to find the drift.

Another approach is the exponentially weighted moving average (EWMA) for concept drift detection. ECDD is one of the drift detection methods and uses EWMA. EWMA finds an increase in mean based on a sequence of random variables Ross et al., 2012. In EWMA, some measures are considered: 1) The probability of not finding the correct classification of an instance before the change point, 2) The standard deviation of the stream. In ECDD, the computation of probability is done online and finds the success or failure probability. The base learner uses the training examples to determine the initial classification accuracy, and the estimators detect the expected time between false positive detections. ECDD works well for gradual drift as compared to sudden drift. The limitations of this method are 1) High computational overhead and 2) Inability to control the false positive rate.

Reactive Drift Detection Method (RDDM) Barros et al., 2017 is based on DDM. RDDM considers some modifications, such as discarding the old instances of very long concepts to detect the drifts as early as possible. It is performed to increase the accuracy of the classifier. The main problem in DDM is that it stays at the warning level for many instances to slow down the incoming instances. It may not detect the existing gradual drift in the data stream.

The variation of the above approach is a fuzzy based drift detection method. FW-DDM is based on the fuzzy-window mechanism for concept drift detection Liu et al., 2017. It is based on a sliding window and contains an overlapping window. In this way, the method finds a more accurate identification of examples from different concepts. The noise can be detected by checking the instance's accuracy of old and current concepts. If there is a degradation in accuracy of both concepts, it is considered as noise.

Farid et al. (2013) build an adaptive ensemble classifier to find concept drift as well as performs data mining and novel class detection. The considered data points belong to the same class that should be closer to each other, but the data points of different classes should be far from each other. This method minimizes the total misclassification error (ERR) in the classification of concept drift.

A similar approach is Learning with Local Drift Detection (LLDD). It monitors the significant change in the error rate of the classification model Gama and Castillo, 2006. Here, it uses a decision tree to determine the drift. A change in the relevant area of the decision tree triggers the drift detection. Thus, it focuses on the change in the local area and reduces the update cost. The approach is based on the VFDT algorithm, where the decision tree split is occurred as per the UFFT algorithm. The error rate of the Naive Bayes classifier is used to classify the nodes while training instances. It defines the drift condition. The limitation of this approach is that the order of examples impacts its performance.

Dynamic Extreme Learning Machine (DELM) uses ELM as classifier Shuliang and Wang, 2017. It differs from OS-ELM by dynamically choosing the number of hidden layer nodes and the different threshold values sets. Another DDM based approach is Drift Detection Method with False Positive rate for multi-label classification (DDM-FP-M) Wang et al., 2020. It takes data from sensors in an IoT environment. Here, the change in accuracy and false positive rate (FPR) is used to detect context change in the data stream.

3.1.2. Sequential analysis based methods:

In this type of concept drift detection algorithm, the data instances are examined sequentially to analyze the change in the data stream context. It signals the drift when the change in data distribution exceeds the specified threshold. The analysis of these types of methods requires a large amount of data examples from the new concept. The following illustrations are based on this approach.

One of the most popular approaches is Page-Hinkley Test (PHT) Page, 1954. It is focused on the change in mean (θ). It holds the distribution of the weighted sum of the last few examples (or observations) to find the moving average of the sample. If the run length (L) is small, the large change in mean (θ) is detected rapidly for small changes. The drift detection takes more time. On the other hand, when L is large, it detects small change efficiently. It is based on hypothesis testing and finds the difference between two samples Mouss et al., 2004.

Ditzler et al. (2015) presents a survey on learning in a non-stationary environment. It discusses active and passive learning approaches used in the non-stationary environment. These change detections approaches are grouped into four categories: 1) Hypothesis testing (HTs), 2) Change-point methods (CPMs), 3) Sequential Hypothesis tests (SHTs), 4) Change detection tests (CDTs). The HTs and CPMs work on a fixed sequence of the data stream. The SHTs focus on the series of incoming data streams up to the decision-making point. The CDTs analyze the statistical behavior of data sequentially.

Similar approaches based on sequential analysis methods are FP-ELM Liu et al., 2016, OD-ELM Sun et al., 2011, and DMDDM, etc. DMDDM has overcome the limitation of cost and execution time Mahdi et al., 2020. It is based on the Page-Hinkley test for drift detection. The limitation of this approach is that it can not work with multi-class classification problems.

3.1.3. Window based methods:

This type of approach accumulates the incoming data instances and forms a batch of data (or a window). Generally, the window based methods contain two windows. The first window is used to store old instances and later has new instances of the data stream. The comparison between these two window instances explained the change in data distribution and signaled the drift. The window size can be fixed or adaptive. A fixed window means that it has the same size of window for the whole analysis. Whereas the adaptive window adjusts the size based on the drift conditions. The data window is shrinked if the drift is detected and expanded when there is no drift condition. The explanation of window based drift detectors is given below. One of the window based methods is Paired Learners. It uses two learners: stable and reactive Bach and Maloof, 2008. The prediction of the stable learner is based on it's experiences. However, the reactive learner predicts the window of current instances. It uses these two learners to analyze the difference in accuracy and to detect the concept drift. The reactive learner has two methods for implementation: Rebuilding the learner based on previous window size instances and a retractable learner that can not learn the instances.

ADWIN is an adaptive sliding window based method. It adapts window dynamically according to the rate of change observed from the data Bifet, 2009. The window size is dynamically increased and shrinked when no change and the change happen in context, respectively. Additionally, ADWIN performs well because it limits the false positives and false negatives rates. The limitation of ADWIN is that it works only for one-dimensional data. It maintains separate windows. Each one belongs to each dimension for n-dimensional raw data. So, it maintains more than one window, and it requires additional time and memory. ADWIN2 is a modified version of ADWIN and requires less time and memory. The drift detection occurs when the average distribution difference of the two consecutive windows is more significant than the predefined threshold. ADWIN2 overcomes the limitation of ADWIN by detecting the slow gradual drift. If the window size is WS, it takes O(log WS) memory and time.

The rate of change (or Volatility Detection) in the stream is discussed in SEED Huang et al., 2014. We already discussed the drift detection, but how frequently the drift arises. It is also an important consideration. SEED determines the cut point for drift detection. It is based on a sliding window mechanism with block size = 3, and the cut point is denoted by the symbol '|'. The consecutive blocks are checked to find homogeneity. If it is present, both the blocks are merged (known as block compression). In this process, it removes the cut points which have low potential. In volatility detection, it finds the interval between the cut points. Here, the cut point means that a point where the change in concept occurs. If there are three intervals, it means no change in volatility, and if it is six, there is a change in volatility. It uses Hoeffding Inequality with Bonferroni correction to find the cut point. SEED performs well in terms of execution time, and in most cases, it requires less memory than ADWIN2.

One of the implicit concept drift detectors is One Class Drift Detector (OCDD) Gözüaçik and Can, 2020, which is based on a sliding window. It approximates the distribution of the new concept. The samples are classified and estimated whether they belong to the current concept or outlier. Here, the outliers percentage is calculated, and the sliding window uses it to signal the drift. It can work with any classifier which has no drift adaptation mechanism.

An ensemble based framework for concept drift detection is present in Shan et al. (2018). It is based on a hybrid labeling strategy that includes long term stable classifier, a dynamic classifier for gradual and sudden drift, on-demand request labeling, dynamically adjusted threshold, etc. The framework uses a multilevel sliding window model and the change in multi-temporal granularity. Whenever the drift occurs, the labeling requirement is increased, but later it stabilizes gradually.

Li et al. (2015) introduces Ensemble decision trees for concept drift (EDTC), a random feature selection variant. These variants perform split-tests and use two Hoeffding bounds inequality with specified thresholds. Random feature selection is made instead of informed split-tests. It builds a random ensemble that is incremental and based on a random decision tree. It dynamically adjusts the drift checkpoint and window size to track the concept drift.

Duda et al. (2018) develops a regression model for data stream. There are many data stream classification methods, but few of them deal with regression problems using heuristic approaches. It models the stationary system under non-stationary noise. It is based on Parzen kernels and Generalized Regression Neural Networks (IGRNN). IGRNN has both weak and strong convergence. These convergences are not found as a property in sliding windows and forgetting mechanism techniques.

Similar approaches based on windowing are DDM Gama et al., 2004, EDDM Baena-Garcıa et al., 2006, ADWIN Bifet and Gavalda, 2007, ADWIN2 Bifet, 2009, STEPD Nishida and Yamauchi, 2007, Paired Learner Bach and Maloof, 2008, FHDDM Pesaranghader and Viktor, 2016, WSTD Roberto and Hidalgo, 2018, Plover algorithm de Mello et al., 2019, etc.

3.1.4. Statistical based methods

Statistical based methods are used to detect the concept drift detection by comparing the distribution of historical and current data instances using statistical tests like Mean, Median, Mode, Kurtosis, Standard Deviation, Regression, Hypothesis Testing, etc.

There are various work done on the basis of statistical test. One of these types of methods is STEPD. Statistical Test of Equal Proportions (STEPD) detects the drift by considering two accuracies: recent and overall Nishida and Yamauchi, 2007. It compares two accuracies. First is the accuracy of a classifier for the current window (with recent examples), and second is the overall accuracy from the beginning of the learning. There is an assumption that both the accuracies are the same. If the target concept is the same as the current concept and there is a significant decrease in recent accuracy, it suggests that the concept is changing. $T(r_l, r_m, n_l, n_m) = \frac{(|r_l/n_l - r_m/n_m| - 0.5(1/n_l + 1/n_m))}{\sqrt{p\prime(1-p\prime)(1/n_l+1/n_m)}}$

Here, r_l is correct classifications among older window (n_l), r_m is number of correct classifications among recent window (n_m), p is significant value, and n is number of instances. A chi-square test is performed for the computation of statistics. Its value is compared with the percentile of the standard normal distribution to obtain the observed significance level. If the value is less than a significance level, the null hypothesis is rejected, and it is assumed that a concept drift occurs. The warning and drift thresholds are also used as in DDM, EDDM, PHT, and ECDD. STEPD reduces misdetection and improves drift detection when the drift is gradual (comparable result with EDDM). But it gives the best outcome for sudden drift.

Dynamic Clustering Forest (DCF) is a statistical based method Song et al., 2016. It deals with concept drift over the data stream. This method constructs several clustering trees (CTs). Each CT represents the different concepts of the data stream. Here, DCF uses two approaches, namely Discriminate CTs and Dual Voting Strategy, which increase classifier's reliability and accuracy. DCF calculates the Mean Square Error (MSE) as a standard measure.

One of Hoeffding's Bounds approach is HDDM Frías-Blanco et al., 2014. This method uses probability inequalities instead of the probability distribution function. It is based on statistical process control. There are two predefined values: warning level at 95 % and drift level at 99%. It has three states STABLE (no change detection), WARNING (possibility of change occurrence), and DRIFT (drift identified). In the warning level, it stores new examples and tries to train the alternative classifier. At the drift level, it adapts the learning model by replacing the old model. This method is independent of the learning algorithm. So, it can apply to any classifier to track the change. HDDM conducts two types of test: $HDDM_{A-test}$ compares the moving averages to detect the drifts and $HDDM_{W-test}$ uses the EMWA forgetting scheme to find the weight of moving average in the data stream. Its time and space complexity is O(1) because it stores relevant information. This method needs to explore with different weighting schemes and applications for real-world problems.

FHDDM is a Fast Hoeffding Drift Detection Method for evolving data streams. The main goal of any drift detector is to minimize the false positives and false negatives. It detects the drift when there is a difference between the maximum probability of correct prediction and the observed current probability of an accurate forecast. It performs the following steps for drift detection. In the first step, it instantiates an object. The second step is to call the DETECT function to detect the drift in the data stream. The final step gives the drift information based on predictions. Each time the old window is removed whenever the window is full. It calculates the number of True Positives (TP), False Positive (FP), and False-Negative (FN) for drift detection. Acceptable delay length is found based on the counting of TP, FP, and FN.

Fisher's Exact test statistical test is used in Fisher Test Drift Detector (FTDD) Rafael, 2018. It finds the changes in the data distribution using a statistical test. The calculation of p-value is different for FTDD that avoids the extreme values in the intermediate results, which should be more precise. For n number of instances, the space and time complexity are O(1) and O(n), respectively.

An approach motivated by Statistical Learning Theory (SLT) is the Plover algorithm. It is an unsupervised concept drift detection algorithm. The method is built to overcome the lack of theoretical foundation problem. It divides the data stream using a windowing mechanism and evaluates mean, variance, skewness, and kurtosis de Mello et al., 2019.

The novel anomaly detection in an unsupervised fashion is discussed in Ahmad et al. (2017). It is based on hierarchical temporal memory (HTM). It continuously learns and models the spatial-temporal characteristics from the input data. It computes three measures: 1) Predictive error, 2) Probabilistic model, 3) Likelihood. The changes in the data stream are detected based on the threshold of likelihood.

The statistical inference is defined in González et al. (2019). It is based on the statistical inference that makes the sequential forecast for the incoming observations. It uses a prequential evaluation, which considers three strategies: 1) Basic window, 2) Sliding window, 3) Fading factor or forgetting mechanism. The method uses these variations in the incoming data and finds out which combination of single mechanisms provides accurate accuracy. It performs statistical tests like Z-test, Nemenyi test, and Friedman method on datasets.

A new stability concept and a change detection algorithm work on unsupervised learning that are explained in Vallim and De Mello (2014). Here, the concept change detection is based on the surrogate data, which is taken as the input. This type of data is part of original data used for evaluation purposes. The surrogate data series is generated from the original series of the data stream. For this purpose, a Fourier transformation is performed that contains some of the original data's characteristics. Therefore, the comparison among the data batches is more accurate than the comparison of consecutive changes in the batches.

3.1.5. Significance analysis based methods

This type of method uses hypotheses analysis for drift detection. These methods are generally used with the window based and statistical analysis based methods. Hypothesis analysis is based on the comparison of the distribution of the old and current data instances. Here, it considers two hypotheses to find the statistical significance: Null hypothesis H_0 and Alternate hypothesis H_a. The statistical hypothesis test is based on statistical inference. For the probability distribution of the data, the alternative hypothesis is defined explicitly. The hypothesis test describes which outcome study may reject the null hypothesis at a pre-specified significance level while using a measure of deviation from the hypothesis. The significance level defines the maximal allowable "false positive rate". Multiple hypotheses also occur in this type of approach. They are classified into two groups: Parallel and Hierarchical multiple hypothesis test based approach.

One of the hypothesis based approach is Wilcoxon Rank Sum Test Drift Detector (WSTD), also known as Mann - Whitney U test Roberto and Hidalgo, 2018. It follows STEPD. It is a non-parametric test. It analyzes two independent samples and determines whether they come from the same distribution. For this method, it is necessary to choose the significance level (α). The normal distribution is considered to choose the criteria for the null hypothesis. Suppose n_1 and n_2 are two samples, and they are combined $(n_1 + n_2)$ in ascending order. The test statistic is calculated with mean and standard deviation to find the value of z. Here, z is defined as $z = (R - \mu_R)/\sigma_R$, and population mean is $\mu_R = n_1 \ * (n_1 + n_2 + 1)/2$, and standard deviation is represented by:

$\sigma_R = \sqrt{n_1 \ * n_2 \ * (n_1 + n_2 + 1)/12}$ where, R is smallest sum of the ranks of both samples, n_1 is size of smallest sample and n_2 represent the size of the largest sample. The z value is used to reject the null hypothesis and the p-value (or obtained probability) also required to find the z value.

Another approach is Fourier Inspired Windows for Concept Drift detection (FIWCD). It examines the buffer window, which is a combination of a larger window and an overlapping window. Discrete Fourier Transform (DFT) is used to find the size of the window. This approach limits the false-positive rate and false-negative rate. The drift is found on the basis of the threshold. The other approaches based on hypothesis testing are: CUMSUM chart Page, 1954, JIT Alippi and Roveri, 2008, DDF Bruno and Santos, 2015, CI-CUMSUM, HHT-CU Yu et al., 2018, TSMSD-FWMA Raza et al., 2015, HLFR Yu and Abraham, 2017, HHT-AG Yu et al., 2018, E-detector Lei et al., 2015, LSDD Li et al., 2017, etc.

JIT builds a just-in-time classifier. It uses PCA for reduction of the feature vector, CUMSUM test for threshold, and CI-CUSUM test for Alternate hypothesis. The approach does not consider any assumptions and theoretical information. Still, it works for multidimensional data. LSDD-Inc is based on the Least-squares density difference. It detects the drift using fixed kernel centers with Chi-square distributions which are non-central and non-independent. Drift Detection Ensemble works parallel with three detectors using the single base classifier. HHT-AG and HHT-CU are based on the Request-and-Reverify scheme. EWMA based method is TSMSD-FWMA, where K-S test and Hotelling T-Squared are used for the univariate and multivariate data stream, respectively.

3.1.6. Data distribution based methods

This type of drift detection method uses old data (or historical data) and current data instances to find the change in context. These types of methods are generally used with the window based approach and analyze statistical significance. The computation of data distribution change provides information about the location where the drift occurs. Therefore, it may lead to computational costs.

One of the data distribution based methods is Hybrid Forest Rad and Haeri, 2019. It focuses on random forests with Hoeffding trees. The method is divided into three stages. The first stage is to create a data frame for obtaining several features. In the second stage, it generates a weak learner. The weak learner has $\sqrt{(f_t)}$ features, where the feature vector of data set is represented by f_t. All the learners are run simultaneously, and after that, there is a combined phase to compute the final result, which is the third stage.

OS-ELMs is a method that builds two OS-ELM models. The first model trains the collected data until a particular time and determines the system's old behavior. Another model is used to update as it gets more information after some time interval. The method identifies the degree of dissimilarity between the models. The dynamically defined thresholds can determine it, and it is necessary to have a bulk amount of data.

The semi-supervised learning based method is STDS. Training set from the labeled data helps to provide the self-training of unlabelled data and predicts their labels. Similarity information is taken from the unlabeled data as prior knowledge. Different clustering techniques are used to label the unlabeled data because the initial classifier could not exploit the information up to a significant level. Here, the Labeling by Ensemble of Clusters (LEC) algorithm is used. Clusters are built using intra-cluster similarity or dissimilarity. Majority voting is performed, and it aggregates the predicted labels to achieve the high confidence pseudo labels.

Other approaches in this category are as follows: PCA based change detection framework (PCA-CD) Qahtan et al., 2015, Statistical Change Detection for multi-dimensional data (SCD) Song et al., 2007, Incremental version of LSDD-CDT Li et al., 2016, LSDD-INC Li et al., 2017 and Local Drift Degree based Density Synchronized

Drift Adaptation (LDD-DSDA) Liu et al., 2017, Sync Stream, Density Estimation (EDE) Feng and Zhang, 2016, Least Squares Density Difference based Change Detection Test (LSDD-CDT), Competence Model based drift detection (CM) Ning et al., 2016, etc.

3.1.7. Decision Boundary based Methods

Decision Boundary based Methods generally form a boundary using initial instances of the data stream. The change in decision boundary is considered as a drift.

Unsupervised learning based algorithm is MD3 Sethi and Kantardzic, 2017. It has different features, like model independent, distribution independent, application independent, etc. It has a region (or margin) based on the best guess. The region learns the information from the data. Classifier builds a margin of a sample, and this margin is set during the training phase. A sudden increase in the sample size within a margin and a change in margin characteristics leads to a change in the data distribution in the testing phase. In this way, it detects a new class in the data stream.

Another method is the Nearest neighbor based density variation identification method (NN-DVI) Liu et al., 2018. It is used to estimate the regional density to detect the concept drift. It has three major components. The first one is the k-nearest neighbor based space partitioning schema (NNPS). It is used to transform the discrete data instances, which are not measurable, into a set of shared subspace. In this way, the density estimation is evaluated. The second component is used to capture the density discrepancies above the subspace and quantify the overall differences. With this component, some density function is built. The last component is the tailored statistical significance test. It can accurately determine the confidence interval of the concept drift. The identification of sudden, gradual, and regional drift is analyzed by this method.

One of the density based clustering approaches is SNDC. It finds the clustering error Shuliang et al., 2020. Here, the neighbor entropy is used to find the similarities. Other density based approach is Re-DBSCAN Miyata and Ishikawa, 2020. It is based on DBSCAN. It is a graph method, and it updates the k-dist graph. Re-DBSCAN detects the origin of drift by updating k-dist graph, which speed-up the learning model revision process.

3.1.8. Model dependent based methods

The model dependent methods compute the change in the distribution of input attributes, and thereby, it tends to change the joint probability distribution $P(\textit{Target Class}|\ \textit{Input Attributes})$. The method creates a learning model using some instances of the data stream and checks the performance of the model for incoming data stream instances. These types of methods use the Kolmogorov-Smirnov test, KL divergence, Wilcoxon rank-sum test, etc.

ExStream is one of the concept drift detection methods which is based on model explanation and uses the attribute-value pair for prediction of outcomes Demšar and Bosnić, 2018. It uses the single-dimensional stream monitoring algorithm, like SPC and PH. It is most suited for parallelization, where each attribute runs separately. IME method is used to classify the stream and observe the dissimilarity between the measures in the stream. There are three variants of this method: 1) ExStreamModel - for an explanation of vector, 2) ExStremAttr - compares the explanation vector using the attribute of n-dimensional stream, 3) ExStreamVal - compares the contribution of the attribute value and ignores the measuring dissimilarities.

Demšar and Bosnić (2018) presents a concept drift detector for labelled data. The method is not dependent on the learning algorithm. The change observation is done based on the change in the model. The computation is based on the attribute-value pairs used for predictions. It describes the decision-making process to create a model and also provides transparency.

Several clusters and micro cluster mechanisms are used to adapt to the change. A new stability algorithm for unsupervised data stream is presented by Vallim and De Mello (2014). The algorithm is based on the generated surrogate data instead of data reordering. It generates a model after processing the sub-series or sub-stream.

Drift detectors address the problems generated due to the change in context and subsequently overcome these problems by adapting the required solutions. It can be done by using several learning approaches like supervised, unsupervised, and semi-supervised techniques. In this way, the concept change is detected in the data stream, and useful knowledge is extracted to update the learning algorithm.Table 3 shows drift detection algorithms with their used techniques. It defines the various learning methods. Table 4 illustrates the chronology of different concept drift detection algorithms. It describes various algorithms and their key points. In addition to this, the advantages and limitations of the algorithms are also present in the table. The comparison of various drift detection algorithms is described in Table 5,6 based on their characteristics, such as finding different types of drifts by the algorithms, algorithms dealing with noisy data, and the existence of window mechanisms. Data sets, repositories, and evaluation measures of concept drift detection algorithms are exhibited in Table 6.

Table 3
Drift detection algorithms with their used techniques

Algorithms	Techniques
Drift Detection Method (DDM)	Perceptron, Neural Network, Decision Tree
Early Drift Detection Method (EDDM)	Decision Tree, Nearest-Neighbourhood Learning Algorithms
Statistical Test of Equal Proportions (STEPD)	IB1, Naive Bayes
Paired Learners (PL)	Naive Bayes, Hoeffding trees, Decision Tree
Adaptive Windowing (ADWIN), ADWIN2	Naive Bayes
Exponentially Weighted Moving Average (EWMA)	Exponentially Weighted Moving Average Chart, KNN
SEED Drift Detector (SEED)	Hoeffding Inequality with Bonferroni Correction
Drift Detection Methods based on Hoeffding's Bounds (HDDM)	Naive Bayes, Perceptron
Fast Hoeffding Drift Detection Method (FHDDM)	Hoeffding Tree, Naive Bayes
Reactive Drift Detection Method (RDDM)	Naive Bayes
ExStream	Page-Hinkley test, Statistical Process Control Algorithm
Fisher Test Drift Detector (FTDD)	Naive Bayes, Hoeffding Tree
Plover algorithm	Markov Inequality, McDiarmid's Inequality
Nearest Neighbor-based Density Variation Identification (NN-DVI)	K-Nearest Neighbor, Naive Bayes, Hoeffding trees
Hybrid Forest	Hoeffding Tree, Random Forest
Online Sequential Extreme Learning Machines (OS-ELMs)	Neural Network, Euclidean distance
Self-Training Data Streams(STDS)	Incremental Naive Bayes learner, Labeling by Ensemble of Cluster (LEC)
Fuzzy Windowing Drift Detection Method (FW-DDM)	Fuzzy set theory
One-Class Drift Detector (OCDD)	Hoeffding Tree, Naive Bayes

Table 4
Chronology of different Concept Drift Detection Algorithms in data stream

Algorithms with Citation	Year	Key points	Advantages	Limitations
PHT Page (1954)	Page, 1954	-An abrupt change detection of the average of a Gaussian signal. -Testing is done between the no-change hypothesis.	-It can detect a small change in data for large run-time efficiently.	-It considers only limited types of drifts.
DDM Gama et al. (2004)	Gama et al., 2004	-It works on the binomial distribution.-It detects abrupt drifts correctly.	-Abrupt and gradual (not very slow) change detection.	-The performance of DDM is usually worsening when the concepts are very large. -In the case of slow-gradual drift, it takes larger space and memory. -It does not support the noisy data stream.
EDDM Baena-Garcıa et al. (2006)	Baena-Garcıa et al., 2006	-It is a distance-based change detection method. -It focuses only on the false positives rate.	-Detect slow-gradual drift. -It works well with noise.	-It is sensitive towards noise and errors. That's why the number of false alarms is high. -The parameters do not set automatically.
STEPD Nishida (2008)	Nishida, 2008	-It considers two accuracies: 1) Recent 2) Overall. -It considers a decrease in accuracy as a drift.	-It can detect sudden drift, gradual drift, and noise. -It gives the best result for sudden drift as compared to DDM, EDDM, PHT, and ECDD.	-Predictive accuracy is worsened by false alarms.-The Hypothesis testing is not appropriate for imbalance, sparse and small size of data streams.
PL Bach and Maloof (2008)	Bach and Maloof, 2008	-It detects accuracy differences between two learners.	-It takes less time and space to get comparable accuracy.	-The performance of the algorithm decreases with an increase in noise. There is less gap between the drift points, leading to an insignificant measurement of the data stream.
ADWIN, ADWIN2 Bifet and Gavalda (2007); Bifet (2009)	Bifet and Gavalda, 2007 Bifet, 2009	-ADWIN2 overcomes the limitation of ADWIN w.r.t. time and memory.	-ADWIN2 is efficient with time and memory.	-It can work only for one-dimensional data.
EWMA Ross et al. (2012)	Ross et al., 2012	-It considers the classification error to detect the drift. -It can control the false positive rate.	-It does not require storage for data in memory and only adds O(1) overhead to the classifier. -There is no storage required for data instances.	-Classification is done only for two classes.
SEQDRIFT Pears et al. (2014)	Pears et al., 2014	-It is based on reservoir approach to improve the detection for slowly varying data.	-It has better false-positive rate than PH, EWMA, and ADWIN.	-Needs to define false positive rate.
SEED Huang et al. (2014)	Huang et al., 2014	-Analyze the rate of change (Volatility Detection) in stream and drift detection concurrently.	-It has a low false-positive rate and lower execution overheads than ADWIN2.	-User-defined threshold. -It does not cover the different types of drift in the data stream.
HDDM Frías-Blanco et al. (2014)	Frías-Blanco et al., 2014	-It uses supervised incremental learning to detect the drift. -It is based on probability inequalities.	-It is independent of the learning algorithm.	-It has poor performance if there is high non-linearity in the classification boundary.
FHDDM Pesaranghader and Viktor (2016)	Pesaranghader and Viktor, 2016	-It observes the difference between the maximum probability of accurate prediction and the currently observed possibility of correct prediction. -It uses a sliding window and Hoeffding's inequality to find the right prediction.	-Highest accuracy and minimum false positives as compared to ADWIN, DDM, HDDM, and EDDM. -Model is distribution independent.	-It can not detect recurrent drift.
RDDM Barros et al. (2017)	Barros et al., 2017	-It periodically minimizes the number of instances of very long stable concepts.	-Accuracy is better than ECDD, STEPD, and DDM. The space and time complexity of most data sets are O(1) and O(n), respectively.	-It Consumes more memory than ECDD, STEPD, and DDM.
FW-DDM Liu et al. (2017)	Liu et al., 2017	- It is fuzzy set theory. -Test statistic is based on DDM.	-Noise can be identified from the data stream.	-It detects only abrupt drift.
MD3 Sethi and Kantardzic (2017)	Sethi and Kantardzic, 2017	-It signals the change when it observes a drop in the classifier's predictive performance.	-Detect concept drift for unlabeled data. -It is not dependent on data distribution.	-Data points consider as a part of the class when they are coming in a particular shape. -More efficient labeling is required.
FTDD Rafael (2018)	Rafael, 2018	-Fisher's Exact test is applied in data mining to discover the dependencies between attributes.	-The space and time complexity for n number of instances are O(1) and O	-The test is expensive in terms of computation.

Journal of King Saud University – Computer and Information Sciences 34 (2022) 9523–9540

Table 4 (*continued*)

Algorithms with Citation	Year	Key points	Advantages	Limitations
			(n), respectively.	
ExStream Demšar and Bosnić (2018)	Demšar and Bosnić, 2018	-It Observes the change in model explanations at different levels.	-It handles redundancies, disjunctions, and noise. -An algorithm is built to act as a classifier even for the non-incremental classifier.	-More parameters, predefined measures, and computation time are required.
WSTD Roberto and Hidalgo (2018)	Roberto and Hidalgo, 2018	-This method contains only two rank values, and the values are present after applying the test.	-It can limit the older window size of STEPD. -It has better performance than SEED, ADWIN, ECDD, STEPD, and SeqDrift2.	-It is a simple method but computationally expensive because of the sorting of observations.
NN-DVI Liu et al. (2018)	Liu et al., 2018	-Concept Drift detection is based on the estimation of regional density.	-NNPS is a new space partition method used to improve the sensitivity of regional drift.	-The performance is limited by the window size. -The parallel distance measure is required to achieve good overall performance.
Plover Algorithm de Mello et al. (2019)	de Mello et al., 2019	-Finds the diverges of the mean from current data to historical data and compares with the threshold. -It stores data in buffer or queue.	-Incremental unsupervised algorithm. -It gives theoretical guarantees for learning.	-Distribution of data should be identical. -Need to set the predefined parameters.
Hybrid Forest Rad and Haeri (2019)	Rad and Haeri, 2019	-The method identifies weak learners in classification or regression tasks.	-It minimizes initial delay as much as possible. -Fast startup is possible.	-More efficient computational resources are needed.
OS-ELMs Yang et al. (2019)	Yang et al., 2019	-The methods are used to identify the new type of drift, such as incremental virtual drift and sudden hybrid drift. -It contains two models, and comparative analysis has been done between models.	-Number of model updates is less.	-User-defined parameters setting is required.
STDS Khezri et al. (2020)	Khezri et al., 2020	-It is a semi-supervised approach. - Kullback-Leibler's (KL) divergence approach is used to find out the concept drift.	-It is a self-training algorithm in a non-stationary environment.	-It may not use all the required information when it deals with unlabeled data in the training procedure.
OCDD Gözüaçik and Can (2020)	Gözüaçik and Can, 2020	-This approach is based on sliding window and one-class classifier.	-It is an implicit drift detection method.	-Adaptive parameterization is not present in the method.

It shows various evaluation measures used in the streaming environment to analyze the performance of the algorithms.

4. Concept drift adaptation

Non-stationary characteristics of streaming data often require a prediction model to adapt the concept drift. Due to the limitation of computational resources like memory and time, we need to predict the patterns in a limited time. The data distribution change occurs in the stream over time because of the concept drift phenomenon, and the labeling in some applications is required for adaptation. The classifier should have adaptation abilities because the data in motion can change over time. Ensemble methods are the most suitable for adaption. The ensemble method amalgamates the multiple classifiers whose predictions are combined to predict the new incoming data instances. It enables the classifier to improve the prediction ability and decomposes the complex problems into multiple sub-problems Krawczyk et al., 2017.

The learning is required to monitor the change in non-stationary distributed environments, and accordingly, it deals with an adaptive mechanism. The adaptive mechanism is necessary for many fields like employment status, change in age, seasonal impact, market behavioral change, demand and supply change, etc. There are two adaptive mechanisms:

- First one is an active approach, in which the data distribution is changed if the drift is detected explicitly.
- Second approach is the passive approach, which updates the model continuously over time Ditzler et al., 2015.

There are two types of drift detection method to handle the concept change: implicit (or blind) and explicit drift detection method. In the implicit method, the decision model updates regularly at a particular time interval. This update is not dependent on the occurrence of concept drift. The disadvantage of this method is that there is wastage of resource consumption, i.e., the model is also updated when the drift does not occur, and it overfits the data for updating the model. The explicit drift detection method observes the data statistics to find the drift. Here, a statistical test is used to find changes in the data. Two common explicit drift detection approaches are: 1) To monitor the error of a base learner, 2) To monitor the data distribution features. In an error based approach, the error may not properly reflect the drift in the concept. Accuracy measure is used for prediction in such approaches. If the training process is not up to the mark, the decision model encounters false alarms (or misdetection). It occurs because the model tries to generalize the problem (or overfitting). For building an adaptive model, the classifier needs to update regularly to get a good prediction. Generally, the adaptive mechanism incorporates a window and forgetting mechanism to build a model up-to-date.

4.1. Classification and Clustering

The classification verifies the current decision model and explains whether the novelty exists in the new example. Classifier categorizes that the particular recent example is a novel class or normal class. Here, the normal class means that the example belongs to the previous samples' class, i.e., no change is incorporated Krawczyk et al., 2017; Farid and Rahman, 2012; Krawczyk et al., 2013. The above approach uses one example, so it may be possible that it does not find the correct decision over different samples at a time. Two different approaches of the classifier are used to classify new examples.

Single classifier: In a single classifier, one classification algorithm is required to build a model. Many classifiers existed, such

Table 5
Comparison of various Concept Drift Detection Algorithms based on their Characteristics

Algorithms	Abrupt Concept Drift	Gradual Concept Drift	Noise	Slow Gradual Concept Drift	Window Mechanism
DDM	✓	✓	×	×	×
EDDM	✓	✓	✓	✓	×
STEPD	✓	✓	✓	✓	✓
PL	✓	✓	✓	✓	✓
ADWIN,ADWIN2	✓	✓	–	✓	✓
EWMA	✓	✓	×	–	✓
SEED	✓	✓	–	✓	✓
HDDM	✓	✓	×	–	✓
FHDDM	✓	✓	✓	–	✓
RDDM	✓	✓	✓	–	×
ExStream	✓	✓	✓	–	✓
FTDD	✓	✓	✓	–	✓
WSTD	✓	✓	✓	–	✓
Plover Algorithm	✓	✓	×	–	✓
NN-DVI	✓	✓	✓	–	✓
Hybrid Forest	✓	✓	×	–	✓
GNG	✓	✓	×	–	✓
OS-ELMs	✓	✓	×	–	✓
FW-DDM	×	✓	✓	–	✓
OCDD	✓	✓	×	–	✓

Table 6
Data set, repository and evaluation measures of concept drift detection algorithms

Algorithms	Data set and Repository	Evaluation measures
DDM	Synthetic datasets (UCI), Electricity Market (UCI), ELEC2 (UCI)	Error rate
ExStream	Airline (UCI), Electricity (UCI)	MTFA, FAC, MTD, MDR, MTR
EDDM	Synthetic datasets (UCI), Electricity Market (UCI), ELEC2 (UCI)	Prequential error
ADWIN, ADWIN2	Electricity Market (UCI)	Rate of false positives, μ
STEPD	STAGGER (UCI), GAUSS (UCI), MIXED2 (UCI), CIRCLES (UCI), HYPERP (UCI)	Prediction error rate
PHT	Process Inspection	Wald test, Average run length
PL	Stagger (UCI), SEA (UCI), Electricity (UCI)	AUC
EWMA	GAUSS (UCI), SINE (UCI), Electricity (UCI), Colonoscopic video sequencing	Accuracy , Standard error
SEQDRIFT2	Rotating hyperplane generator (UCI), Airline (UCI), Poker hand (UCI)	False positive rates, Correction factor
SEED	Sensor Stream (http://www.cse.fau.edu), Forest Covertype (MOA)	FPR, TPR, Delay, Execution time, Run-time, Memory Usage, Accuracy
HDDM	Electricity Market (MOA), Forest cover type (MOA), USENET1 (MOA)	Mean, Accuracy
FHDDM	Synthetic Datasets (UCI), Airlines (MOA), Poker Hand (MOA), Electricity	Drift detection delay, TPR, FPR, FNR
FTDD	Synthetic Datasets (UCI), Airlines (MOA), Poker Hand (MOA), Cover type	Precision, Recall, and F-Measure (F1)
WSTD	Airlines (MOA), Connect4 (UCI), Covertype (MOA), Spam (UCI), Outdoor (https://github.com/vlosing/driftDatasets/), Pokerhand (MOA)	Accuracy
Plover algorithm	Historical prices of Bitcoin (https://coinmarketcap.com/currencies/bitcoin/historical-data/?start=20130428 and end=20180831), Electricity (MOA), Airline (MOA)	Diverges (div), Dynamic time warping
NN-DVI	Electricity (MOA), Weather (http://users.rowan.edu/polikar/research/NSE), Spam filtering (UCI), Twenty Newsgroups (UCI)	Average Rank, Accuracy
Hybrid Forest	Wind (UCI), Abalone (UCI)	Accuracy, $P_{confidence}$ $p_{notobserved}$
OS-ELMs	Electricity (MOA), The Nebraska weather (http://users.rowan.edu/ polikar/nse.html)	$Gain_{RMSE}, Gain_{MAE}$
STDS	SEA (MOA), Rotating Hyperplane (MOA), Electricity (UCI), Forest Cover-Type (UCI)	Accuracy
MD3	Spam (UCI), Spamassassin (UCI), Phishing (UCI) , Nsl kdd (UCI)	AccTr, $\theta_{margin-of-uncertainty}$
FW-DDM	SEA, Electricity, Airline, Spam	Accuracy, Precision, Recall, F1, Time (ms)
OCDD	Electricity, Airline, Spam, Covtype, Pokerhand, Outdoor, Rialto, Phissing, Rotating hyperplane, Mixed, Moving Squares, Moving RBF, Interchanging RBF (https://github.com/ogozuacik/one-class-drift-detection)	Accuracy

as SVM, ANN, Naïve Bayes, decision model, KNN, etc. In the case of the decision model, it is composed of three categories, namely normal, extension, and novelty sub-models. Every sub-model is analyzed. The example is assigned to a class that is associated with the cluster's sub-model. There is only one decision model without sub-models, which is composed of clusters. The clusters learn the data samples offline and online. The cluster has the closest distance used to classify a new example. The new example is receiving a label associated with a cluster. In this case, the new examples are classified in one of the existing learned classes in the online or new class evolution phase.

Ensemble classifier: It is used to classify a new example that has an unknown label Masud et al., 2011; Song et al., 2016. SEED, ADWIN, PH, CUSUM1, CUSUM2, $HDDM_a, HDDM_w$, EDDM, DDM and EWMA are ensemble approaches. The classifiers verify whether the example is unknown. It generally checks with the use of the decision tree of the ensemble. When the example does not belong to any of the clusters, it will put an example on the leaf node. Classifier ensemble marks as unknown. Otherwise, it is classified by the ensemble of majority voting method Farid et al., 2013.

Clustering is the process of making a group of data into classes of similar data. It keeps similar data instances into a single cluster. It arranges the data samples to find the concept drift, outliers, or new classes. It is used to build a model represented by a set of clusters and classifies the unknown label options Faria et al., 2013. In real-time scenarios, unsupervised and semi-supervised learning are the most attractive research area due to the low labeling of data. In unsupervised learning, the clustering method is used to manage the data instances in a cluster form. Every sample is categorized into a group based on distance. The distance between clusters is calculated. Suppose the distance is less than the radius of the cluster. The decision model explains the example; otherwise, it is marked as unknown Spinosa et al., 2009.

Other researchers use various algorithms for classification. In KNN, if an example belongs to the nearest cluster or receives the highest frequency from the cluster, it assigns to that cluster Adriana Sayuri Iwashita, 2019; Shan et al., 2018. In the K-means algorithm, K defines the number of clusters. Each data point assigns based on the similarity of a particular data point with clusters. Finally, if the example does not belong to any class, it marks as unknown because it can be noise, outlier, or new class evolution. Three important aspects are considered in this ensemble approach:

- Update is carried out with (or without) feedback mechanism.
- Number of classifiers used for learning.
- Consideration of forgetting mechanisms to remove old concepts information.

Faithfull et al. (2019) build an ensemble of uni-variate change detectors over multivariate data. Uni-variate detectors consist of only three detectors, and above that is known as a multivariate ensemble. Here, it uses the ensemble of classifiers to detect the concept drift. An ensemble of uni-variate detectors is known as a subspace ensemble. It assumes that the ensemble approach is better than pure unsupervised multivariate techniques. Most multivariate change detection methods require two components: 1) To estimate the incoming data instances distribution, 2) To evaluate whether the new data points fit in the model. Here, either the clustering or multivariate distribution modeling commonly performs the estimation of the streaming data distribution. It uses individual detectors like Hotelling, KL, and SPLL. The approaches like sequential analysis, control charts, and monitoring distributions are used to detect a change in the data stream. In sequential analysis Page, 1954, it considers sequences of examples: $X = [x_1, \ldots, x_n]$. It uses hypothesis testing where H_0 denotes the null hypothesis, and $p_0(x)$ is a probability distribution from which X is generated. H_1 is alternate hypothesis in which X is generated by $p_1(x)$ distribution. After that, the likelihood ratio of two distributions is calculated. It uses two thresholds, namely α and β, which check the target error. The drift is signaled when the change increases from a fixed threshold (λ). It also calculates the positive and negative shifts of data.

Control charts (CC) depend on Statistical Process Control (SPC). The CC is used by many previous papers like Gama et al. (2004); Baena-Garcıa et al. (2006), Ross et al. (2012) and Frías-Blanco et al. (2014). It monitors the classification errors. Bernoulli random variable can be used to calculate the error with the probability of success p. Here, success denotes the occurrence of errors. The probability p is unknown at the start of monitoring. When a new example comes into existence, it is re-estimated, and a portion of errors is encountered.

Monitoring distributions monitor two windows of distribution. The old window has old data, and another window has new data. A statistical test requires to compare these two windows and further apply the null hypothesis. By this process, it detects the point where the drift occurs. The Bifet (2009); Huang et al. (2014), Albert Bifet (2009), etc., use the same mechanism. Adaptive learning algorithms address many issues of data stream mining as given below Khamassi et al., 2018:

- How to track concept drift.
- How to incorporate the forgetting mechanism and when to discard the data instances or which data instances need to replace by the current data instances.
- How adaptive mechanism is used to update the information.
- How to adapt the learner parameters and structure to react according to these new environmental requirements.

4.2. Forgetting mechanism

Data instances are changing over time due to distribution changes. Here, the model is updated using the forgetting mechanism where it forgets the instances that belong to the old concepts. The approaches are based on ensemble (i.e., a combination of more than one classifier) and a single classifier. Generally, a method finds the change in concept. If concepts are new or not seen previously, it removes these concepts and their related information. After that, it replaces the old learning model because of the inefficiency of finding the correct pattern in the current data stream.

The forgetting mechanism is adopted by MINAS, EWMA, and other algorithms. They incorporate it during concept drift detection. When the example does not arrive for a long time that belongs to a previously defined cluster, the cluster is removed to accommodate the new instances in a new evolving cluster. During the concept evolution, the data is stored in the buffer storage, and the old data is stored in sleep memory till further progress. The incremental forgetting mechanism maintains the weights of the old chunk every time and reduces the memory space by removing the old data instances. After reading some chunks, the weight '0' is set to the old examples and provides space for the new incoming examples. Finally, it removes the data information by assuming that they are not crucial for the current stream.

The evolving learning models can be considered as an extension of incremental algorithms. The evolving model of the data distributions is unexplored so far. These types of methods are naturally able to handle the drifts. When the drifts lead to such situations, the model structure is evolved, and a smooth transition from an older model to a newer one can be achieved Wang et al., 2018.

4.3. Windowing mechanism

In the real-time environment, the data stream is continuous and stored in the windows for analysis purposes. The size of the window depends on the applications in which it is used. Some researchers use a static window mechanism to detect a change in context, i.e., each window has a fixed number of instances. Another type of window is a sliding window where the size of the window is expanded and shrinked based on the scenarios. In the sliding window mechanism, the window stores the most recent data instances. The model analyzes the current data instances and updates itself using instances. Whenever the drift is detected, it removes the old data from the window.

The size of the sliding window can vary with full memory, fixed size, and variable (or adaptive) size. It is generally used with different applications or tested the behavior of data stream using varying size windows. In full memory, the data instances are stored in memory until the memory gets exhausted. After that, the old data instances are removed to accommodate the new data instances in free memory space. The size of the window is predefined in the fixed-size window mechanism to store the data. Adaptive window mechanism automatically grows the window whenever no change is seen, i.e., the concept is stable and shrinks the window when there is a change in data distribution Bifet and Gavalda, 2007; Bifet, 2009.

5. Research trends

In the streaming environment, change detection and adaptation are challenging tasks. There are various domains related to data stream mining where the research is focused on concept drift phenomena.

5.1. Drift mapping

In data stream mining, the mapping determines the nature of drift for a specific period. Whereas, in concept drift, the drift detection algorithms try to identify the drift in its occurrence point. Drift mapping is a standalone data analysis task, whereas the drift detector uses an online learning algorithm. Drift mapping analyzes the following scenarios:

- How the different drift mechanisms perform in various forms of concept drift,
- How to find the drift,
- How fast the detector detects the change point.

Drift mapping describes the different types of drifts that occur in some kind of domain. It reveals the form of the drift that is likely to happen shortly.

5.2. Concept evolution

Concept evolution defines the data instances which are not learned by the classifier previously. It describes the new concept data instances. So, the classifier is able to recognize the data instances for further model adaptation. For example, in the network intrusion detection scenario, the class belongs to different types of attacks, and during the computation, the new distribution data instances come into existence. There are limitations in the number of data instances stored in the buffer space until it reaches the specific predefined limit to consider these new instances as a new class. If the data instances qualify as a class, they are considered as a new class, and this mechanism is known as concept evolution in the data stream.

Here, a general approach of concept evolution is described in supervised learning with non-stationary data distribution. Initially, it takes some data points to train a model. It is commonly referred to as the warm-up or offline phase. Further, the data classification is performed using new incoming data during the online learning phase. In this way, the whole process of classification and model building is done. In the concept evolution, it is assumed that the data comes in the form of chunks and receives their true label. Finally, the model is updated. Before the declaration of novel class from the subset of chunks, some conditions should be fulfilled:

- The cohesion between data points should belong to the same class. Same class data points are more similar to the instance of other known classes Masud et al., 2011; Faria et al., 2016; Farid and Rahman, 2012.
- The separation between data points does not belong to the same class. Some algorithms use the slack space around the cluster to build the proper separation between the clusters Masud et al., 2011; Masud et al., 2010.
- The number of instances in the emerging class subset is less than the pre-specified threshold. Such a class is considered an outlier.
- If the number of instances in the emerging class subset is more than or equal to the threshold and belongs to the subspace of hyperspace, it is considered as a novel class.

The above points define some considerations during the novel class detection. However, no fixed framework is defined for the number of data points to declare a class as a novel class. Generally, it takes several data instances which are currently present in data classes. The method cannot provide such restrictions to declare a class as a novel class in real-world scenarios. Sometimes, an important class cannot be recognized as a novel class because it does not have optimum data points.

5.3. Concept shift

The context or concept change occurs due to the underlying data generator changes. So, whenever the change moves from one context to another, it is defined as a conceptual shift. For example, the middle class is a concept in a particular geographical area, and some people migrate from one place to another place due to some reason. In this scenario, the new geographical area also has the same concept of the middle class, but their meaning changes over time because the middle class changes their underlying characteristics. The middle class concept name is similar to the new concept, i.e., middle class, but their meaning is different. Another example is antibiotic resistance, where the medical category of people changes from antibiotic non-resistance to resistance. The transition between these two situations is known as a conceptual shift.

5.4. Volatility detection

There are many drift detectors focused on finding the concept change in the data stream. Still, there is much hidden information in the data stream. So, there is a possibility to analyze the rate of change in the data stream. The rate of change (or volatility detection) defines that there is any rate shift in the data stream. Volatility detection performs simultaneously with the concept drift detector Huang et al., 2014.

There are many fields related to the concept drift in data stream mining, such as multi-dimensional data stream, the variable shape of clusters, identifying the drift shift or concept drift, co-variance analysis, change in the data stream rate, multi-class classification, etc. Further, the optimization can be applied in the concept drift detection methods Abualigah, 2019; Abualigah et al., 2021; Laith Abualigah et al., 2021.

6. Research challenges

Data stream mining is one of the challenges in real-time scenarios. During mining, we have to deal with several issues like new concept generation, change in concept, concept shift, the adaptation of the concept, etc. So, several challenges exist in data stream mining:

- Drift detector should deal with the data streams having features like numeric, categorical, multi-categorical, temporal, binary, and skewness.
- Scalability is a significant concern in data stream mining because we have to build an algorithm to handle a large volume of data with varying velocities.
- Data has many features. The analysis of features that have more impact on the change in concept is a challenging task.
- In the real-time analysis, the analysis of unsupervised and semi-supervised data streams is still a challenge.
- Drift detection algorithm should be able to deal with noise, missing values, class imbalance, and sizeable data stream size.

- Continuous flow of data processing should be done near real-time with limited resources such as CPU, main memory, and cost.
- Windowing and forgetting mechanism are the major challenges in data stream mining because the window size is not stable for all the cases. Another aspect is to analyze the data which are more important or meaningful for learners shortly.

7. Conclusion

The concept evolution, novelty detection, the data stream of infinite length, etc., are major challenges in the streaming environment. There are many drift detectors built to detect concept change. The majority of detectors are based on posterior distribution, error rate change, threshold, etc. The concept change detection methods suffer from many performance factors. These factors are slow adaptation towards the drift, poor sensitivity to gradual drift, high false rate, high computational complexity, and delay in detecting the different types of drifts. Novelty detection and concept evolution are other major concerns in data stream mining. Novelty detectors are either based on one-class classification or multi-class classification problems. Still, many algorithms are based on one-class classification only. Much exploration is needed for the multi-class classification approach because complex analysis is required for multiple classes. The shape of clusters is another constraint because it restricts the data instances to accommodate in the class or outside the class. These classes may miss some data points which have some useful information, and the other condition may arise where two classes identify as single classes. So, these areas still have some challenges that need to be incorporated during data stream mining.

8. Future work

Future research can contribute to several issues of data stream mining:

- There is a need to optimize the false positive rate (FPR) and false negative rate (FNR) to maintain the accuracy of the classifier while detecting the drift.
- The detection of drift needs to converge as close as possible from the actual drift point.
- Multi-dimensional data constitutes a significant concern while detecting the drift.
- Data imbalance problem is still a significant issue in the data stream mining during the analysis of the concept drift.

References

Daniel Kifer, Shai Ben-David, and Johannes Gehrke. Detecting change in data streams. In Proceedings of the Thirtieth international conference on Very large data bases-Volume 30, pages 180–191. VLDB Endowment, 2004.

Gama, Joao, 2010. Knowledge discovery from data streams. Chapman and Hall/CRC.

Khamassi, Imen, Sayed-Mouchaweh, Moamar, Hammami, Moez, Ghédira, Khaled, 2018. Discussion and review on evolving data streams and concept drift adapting. Evolving systems 9 (1), 1–23.

Gert Cauwenberghs and Tomaso Poggio. Incremental and decremental support vector machine learning. In Advances in neural information processing systems, pages 409–415, 2001.

Rasoul Safavian, S., Landgrebe, David, 1991. A survey of decision tree classifier methodology. IEEE transactions on systems, man, and cybernetics 21 (3), 660–674.

Basheer, Imad A, Hajmeer, Ma.ha., 2000. Artificial neural networks: fundamentals, computing, design, and application. Journal of microbiological methods 43 (1), 3–31.

Hart, Peter, 1968. The condensed nearest neighbor rule (corresp.). IEEE transactions on information theory 14 (3), 515–516.

Barros, Roberto Souto Maior, Silas Garrido, T., Santos, Carvalho, 2018. A large-scale comparison of concept drift detectors. Information Sciences 451, 348–370.

Ryan Hoens, T., Polikar, Robi, Chawla, Nitesh V, 2012. Learning from streaming data with concept drift and imbalance: an overview. Progress. Artificial Intelligence 1 (1), 89–101.

Gama, João, Žliobaitė, Indrė, Bifet, Albert, Pechenizkiy, Mykola, Bouchachia, Abdelhamid, 2014. A survey on concept drift adaptation. ACM computing surveys (CSUR) 46 (4), 44.

Demšar, Jaka, Bosnić, Zoran, 2018. Detecting concept drift in data streams using model explanation. Expert Systems with Applications 92, 546–559.

Wang, Shenghui, Schlobach, Stefan, Klein, Michel, 2010. In: What is concept drift and how to measure it? In International Conference on Knowledge Engineering and Knowledge Management. Springer, pp. 241–256.

Geoffrey I Webb, Roy Hyde, Hong Cao, Hai Long Nguyen, and Francois Petitjean. Characterizing concept drift. Data Mining and Knowledge Discovery, 30 (4): 964–994, 2016.

Ahmad, Subutai, Lavin, Alexander, Purdy, Scott, Agha, Zu.ha., 2017. Unsupervised real-time anomaly detection for streaming data. Neurocomputing 262, 134–147.

Juan I González Hidalgo, Bruno IF Maciel, and Roberto SM Barros. Experimenting with prequential variations for data stream learning evaluation. Computational Intelligence, 35 (4): 670–692, 2019.

Ditzler, Gregory, Polikar, Robi, 2012. Incremental learning of concept drift from streaming imbalanced data. IEEE transactions on knowledge and data engineering 25 (10), 2283–2301.

Sebastiao, Raquel, Gama, Joao, 2009. A study on change detection methods. In: In Progress in Artificial Intelligence, 14th Portuguese Conference on Artificial Intelligence EPIA, pp. 12–15.

Adriana Sayuri Iwashita, 2019. Victor Hugo C de Albuquerque, and João Paulo Papa. Learning concept drift with ensembles of optimum-path forest-based classifiers. Future Generation Computer Systems 95, 198–211.

Elwell, Ryan, Polikar, Robi, 2011. Incremental learning of concept drift in nonstationary environments. IEEE Transactions on Neural Networks 22 (10), 1517–1531.

Masud, Mohammad M, Al-Khateeb, Tahseen M, Khan, Latifur, Aggarwal, Charu, 2011. Jing Gao, Jiawei Han, and Bhavani Thuraisingham. Detecting recurring and novel classes in concept-drifting data streams. In: In 2011 IEEE 11th International Conference on Data Mining IEEE, pp. 1176–1181.

Jie, Lu., Liu, Anjin, Song, Yiliao, Zhang, Guangquan, 2020. Data-driven decision support under concept drift in streamed big data. Complex & Intelligent Systems 6 (1), 157–163.

Faria, Elaine R, Gonçalves, Isabel JCR, de Carvalho, André CPLF, Gama, João, 2016. Novelty detection in data streams. Artificial Intelligence Review 45 (2), 235–269.

Gama, Joao, Medas, Pedro, Castillo, Gladys, Rodrigues, Pedro, 2004. In: Learning with drift detection In Brazilian symposium on artificial intelligence. Springer, pp. 286–295.

Nishida, Kyosuke, 2008. Learning and detecting concept drift. Information Science and Technology.

Wang, Shenghui, Schlobach, Stefan, Klein, Michel, 2011. Concept drift and how to identify it. Web Semantics: Science, Services and Agents on the World Wide Web 9 (3), 247–265.

Ramírez-Gallego, Sergio, Krawczyk, Bartosz, García, Salvador, Woźniak, Michał, Herrera, Francisco, 2017. A survey on data preprocessing for data stream mining: Current status and future directions. Neurocomputing 239, 39–57.

Nguyen, Hai-Long, Woon, Yew-Kwong, Ng, Wee-Keong, 2015. A survey on data stream clustering and classification. Knowledge and information systems 45 (3), 535–569.

Silva, Jonathan A, Faria, Elaine R, Barros, Rodrigo C, Hruschka, Eduardo R, De Carvalho, Andre CPLF, Gama, João, 2013. Data stream clustering: A survey. ACM Computing Surveys (CSUR) 46 (1), 13.

Li, Cheng-Te, Shan, Man-Kwan, Jheng, Shih-Hong, Chou, Kuan-Ching, 2016. Exploiting concept drift to predict popularity of social multimedia in microblogs. Information Sciences 339, 310–331.

Manuel Baena-Garcia, José del Campo-Ávila, Raúl Fidalgo, Albert Bifet, R Gavalda, and R Morales-Bueno. Early drift detection method. In Fourth international workshop on knowledge discovery from data streams, volume 6, pages 77–86, 2006.

Ross, Gordon J, Adams, Niall M, Tasoulis, Dimitris K, Hand, David J, 2012. Exponentially weighted moving average charts for detecting concept drift. Pattern recognition letters 33 (2), 191–198.

Roberto SM Barros, Danilo RL Cabral, Paulo M Gonçalves Jr, and Silas GTC Santos. Rddm: Reactive drift detection method. Expert Systems with Applications, 90: 344–355, 2017.

Liu, Anjin, Zhang, Guangquan, Jie, Lu., 2017. Fuzzy time windowing for gradual concept drift adaptation. In: In 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE) IEEE, pp. 1–6.

Dewan Md Farid, Li Zhang, Alamgir Hossain, Chowdhury Mofizur Rahman, Rebecca Strachan, Graham Sexton, and Keshav Dahal. An adaptive ensemble classifier for mining concept drifting data streams. Expert Systems with Applications, 40 (15): 5895–5906, 2013.

Gama, Joao, Castillo, Gladys, 2006. In: Learning with local drift detection In International Conference on Advanced Data Mining and Applications. Springer, pp. 42–55.

Shuliang, Xu., Wang, Junhong, 2017. Dynamic extreme learning machine for data stream classification. Neurocomputing 238, 433–449.

Wang, Pingfan, Jin, Nanlin, Fehringer, Gerhard, 2020. Concept drift detection with false positive rate for multi-label classification in iot data stream. In: In 2020 International Conference on UK-China Emerging Technologies (UCET) IEEE, pp. 1–4.
Page, Ewan S, 1954. Continuous inspection schemes. Biometrika 41 (1/2), 100–115.
Hayet Mouss, D Mouss, N Mouss, and L Sefouhi. Test of page-hinckley, an approach for fault detection in an agro-alimentary production system. In 2004 5th Asian Control Conference (IEEE Cat. No. 04EX904), volume 2, pages 815–818. IEEE, 2004.
Ditzler, Gregory, Roveri, Manuel, Alippi, Cesare, Polikar, Robi, 2015. Learning in nonstationary environments: A survey. IEEE Computational Intelligence Magazine 10 (4), 12–25.
Liu, Dong, YouXi, Wu., Jiang, He, 2016. Fp-elm: An online sequential learning algorithm for dealing with concept drift. Neurocomputing 207, 322–334.
Sun, Yongjiao, Yuan, Ye, Wang, Guoren, 2011. An os-elm based distributed ensemble classification framework in p2p networks. Neurocomputing 74 (16), 2438–2443.
Mahdi, Osama A, Pardede, Eric, Ali, Nawfal, Cao, Jinli, 2020. Diversity measure as a new drift detection method in data streaming. Knowledge-Based Systems 191, 105227.
Bach, Stephen H, Maloof, Marcus A, 2008. Paired learners for concept drift. In: In 2008 Eighth IEEE International Conference on Data Mining IEEE, pp. 23–32.
Bifet, Albert, 2009. Adaptive learning and mining for data streams and frequent patterns. ACM SIGKDD Explorations Newsletter 11 (1), 55–56.
David Tse Jung Huang, Yun Sing Koh, Gillian Dobbie, and Russel Pears. Detecting volatility shift in data streams. In 2014 IEEE International Conference on Data Mining, pages 863–868. IEEE, 2014.
Gözüaçik, Ömer, Can, Fazli, 2020. Concept learning using one-class classifiers for implicit drift detection in evolving data streams. Artificial Intelligence Review, 1–23.
Shan, Jicheng, Zhang, Hang, Liu, Weike, Liu, Qingbao, 2018. Online active learning ensemble framework for drifted data streams. IEEE transactions on neural networks and learning systems 99, 1–13.
Li, Peipei, Xindong, Wu., Xuegang, Hu., Wang, Hao, 2015. Learning concept-drifting data streams with random ensemble decision trees. Neurocomputing 166, 68–83.
Duda, Piotr, Jaworski, Maciej, Rutkowski, Leszek, 2018. Convergent time-varying regression models for data streams: Tracking concept drift by the recursive parzen-based generalized regression neural networks. International journal of neural systems 28 (02), 1750048.
Bifet, Albert, Gavalda, Ricard, 2007. Learning from time-changing data with adaptive windowing. In: In Proceedings of the 2007 SIAM international conference on data mining SIAM, pp. 443–448.
Nishida, Kyosuke, Yamauchi, Koichiro, 2007. In: Detecting concept drift using statistical testing In International conference on discovery science. Springer, pp. 264–269.
Pesaranghader, Ali, Viktor, Herna L, 2016. Fast hoeffding drift detection method for evolving data streams. In: Joint European conference on machine learning and knowledge discovery in databases. Springer, pp. 96–111.
Roberto Souto Maior de Barros, Juan Isidro González Hidalgo, and Danilo Rafael de Lima Cabral. Wilcoxon rank sum test drift detector. Neurocomputing, 275: 1954–1963, 2018.
de Mello, Rodrigo F, Vaz, Yule, Grossi, Carlos H, Bifet, Albert, 2019. On learning guarantees to unsupervised concept drift detection on data streams. Expert Systems with Applications 117, 90–102.
Ge Song, Yunming Ye, Haijun Zhang, Xiaofei Xu, Raymond YK Lau, and Feng Liu. Dynamic clustering forest: an ensemble framework to efficiently classify textual data stream with concept drift. Information Sciences, 357: 125–143, 2016.
Frías-Blanco, Isvani, del Campo-Ávila, José, Ramos-Jimenez, Gonzalo, Morales-Bueno, Rafael, Ortiz-Díaz, Agustín, Caballero-Mota, Yailé, 2014. Online and non-parametric drift detection methods based on hoeffding's bounds. IEEE Transactions on Knowledge and Data Engineering 27 (3), 810–823.
Danilo Rafael de Lima Cabral and Roberto Souto Maior de Barros. Concept drift detection based on fisher's exact test. Information Sciences, 442: 220–234, 2018.
Vallim, Rosane MM, De Mello, Rodrigo F, 2014. Proposal of a new stability concept to detect changes in unsupervised data streams. Expert Systems with Applications 41 (16), 7350–7360.
Alippi, Cesare, Roveri, Manuel, 2008. Just-in-time adaptive classifiers–part ii: Designing the classifier. IEEE Transactions on Neural Networks 19 (12), 2053–2064.
Bruno Iran Ferreira Maciel, Silas Garrido Teixeira Carvalho Santos, and Roberto Souto Maior Barros. A lightweight concept drift detection ensemble. In 2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI), pages 1061–1068. IEEE, 2015.
Shujian Yu, Xiaoyang Wang, and José C Príncipe. Request-and-reverify: Hierarchical hypothesis testing for concept drift detection with expensive labels. arXiv preprint arXiv:1806.10131, 2018.
Raza, Haider, Prasad, Girijesh, Li, Yu.hua., 2015. Ewma model based shift-detection methods for detecting covariate shifts in non-stationary environments. Pattern Recognition 48 (3), 659–669.
Shujian Yu and Zubin Abraham. Concept drift detection with hierarchical hypothesis testing. In Proceedings of the 2017 SIAM International Conference on Data Mining, pages 768–776. SIAM, 2017.
Laith Abualigah, Ali Diabat, Seyedali Mirjalili, Mohamed Abd Elaziz, Amir, H Gandomi., 2021. The arithmetic optimization algorithm. Computer methods in applied mechanics and engineering 376, 13609.
Lei, Du., Song, Qinbao, Zhu, Lei, Zhu, Xiaoyan, 2015. A selective detector ensemble for concept drift detection. The Computer Journal 58 (3), 457–471.
Li, Bu., Zhao, Dongbin, Alippi, Cesare, 2017. An incremental change detection test based on density difference estimation. IEEE Transactions on Systems, Man, and Cybernetics: Systems 47 (10), 2714–2726.
Radin Hamidi Rad and Maryam Amir Haeri. Hybrid forest: A concept drift aware data stream mining algorithm. arXiv preprint arXiv:1902.03609, 2019.
Qahtan, Abdulhakim A, Alharbi, Basma, Wang, Suojin, Zhang, Xiangliang, 2015. A pca-based change detection framework for multidimensional data streams: Change detection in multidimensional data streams. In: In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 935–944.
Pears, Russel, Sakthithasan, Sripirakas, Koh, Yun Sing, 2014. Detecting concept change in dynamic data streams. Machine Learning 97 (3), 259–293.
Tegjyot Singh Sethi and Mehmed Kantardzic. On the reliable detection of concept drift from streaming unlabeled data. Expert Systems with Applications, 82: 77–99, 2017.
Liu, Anjin, Jie, Lu., Liu, Feng, Zhang, Guangquan, 2018. Accumulating regional density dissimilarity for concept drift detection in data streams. Pattern Recognition 76, 256–272.
Yang, Zhe, Al-Dahidi, Sameer, Baraldi, Piero, Zio, Enrico, Montelatici, Lorenzo, 2019. A novel concept drift detection method for incremental learning in nonstationary environments. In: IEEE transactions on neural networks and learning systems.
Khezri, Shirin, Tanha, Jafar, Ahmadi, Ali, Sharifi, Arash, 2020. Stds: self-training data streams for mining limited labeled data in non-stationary environment. Applied Intelligence, 1–20.
Song, Xiuyao, Mingxi, Wu., Jermaine, Christopher, Ranka, Sanjay, 2007. Statistical change detection for multi-dimensional data. In: In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 667–676.
Li, Bu., Alippi, Cesare, Zhao, Dongbin, 2016. A pdf-free change detection test based on density difference estimation. IEEE transactions on neural networks and learning systems 29 (2), 324–334.
Liu, Anjin, Song, Yiliao, Zhang, Guangquan, Jie, Lu., 2017. Regional concept drift detection and density synchronized drift adaptation. In: In IJCAI International Joint Conference on Artificial Intelligence.
Feng, Gu., Zhang, Guangquan, 2016. Jie Lu, and Chin-Teng Lin. Concept drift detection based on equal density estimation. In: In 2016 International Joint Conference on Neural Networks (IJCNN) IEEE, pp. 24–30.
Ning, Lu., Jie, Lu., Zhang, Guangquan, Mantaras, Ramon Lopez De, 2016. A concept drift-tolerant case-base editing technique. Artificial Intelligence 230, 108–133.
Shuliang, Xu., Feng, Lin, Liu, Shenglan, Qiao, Hong, 2020. Self-adaption neighborhood density clustering method for mixed data stream with concept drift. Engineering Applications of Artificial Intelligence 89, 103451.
Miyata, Yasushi, Ishikawa, Hiroshi, 2020. Concept drift detection on stream data for revising dbscan. Electronics and Communications in Japan.
Krawczyk, Bartosz, Minku, Leandro L, Gama, João, 2017. Jerzy Stefanowski, and MichałWoźniak. Ensemble learning for data stream analysis: A survey. Information Fusion 37, 132–156.
Dewan Md Farid and Chowdhury Mofizur Rahman, 2012. Novel class detection in concept-drifting data stream mining employing decision tree. In: In 2012 7th International Conference on Electrical and Computer Engineering IEEE, pp. 630–633.
Bartosz Krawczyk and MichałWoźniak. Incremental learning and forgetting in one-class classifiers for data streams. In Proceedings of the 8th International Conference on Computer Recognition Systems CORES 2013, pages 319–328. Springer, 2013.
Faria, Elaine R, Gama, João, Carvalho, André CPLF, 2013. Novelty detection algorithm for data streams multi-class problems. In: In Proceedings of the 28th annual ACM symposium on applied computing ACM, pp. 795–800.
Eduardo J Spinosa, André Ponce de Leon de Carvalho, João Gama, et al. Novelty detection with application to data streams. Intelligent Data Analysis, 13 (3): 405–422, 2009.
Faithfull, William J, Rodríguez, Juan J, Kuncheva, Ludmila I, 2019. Combining univariate approaches for ensemble change detection in multivariate data. Information Fusion 45, 202–214.
Albert Bifet and Richard Kirkby. Data stream mining a practical approach. 2009.
Shuo Wang, Leandro L Minku, and Xin Yao. A systematic study of online class imbalance learning with concept drift. IEEE transactions on neural networks and learning systems, (99): 1–20, 2018.
Masud, Mohammad, Gao, Jing, Khan, Latifur, Han, Jiawei, Thuraisingham, Bhavani M, 2010. Classification and novel class detection in concept-drifting data streams under time constraints. IEEE Transactions on Knowledge and Data Engineering 23 (6), 859–874.
Abualigah, Laith Mohammad Qasim et al., 2019. Feature selection and enhanced krill herd algorithm for text document clustering. Springer.
Abualigah, Laith, Yousri, Dalia, Elaziz, Mohamed Abd, Ewees, Ahmed A, Al-qaness, Mohammed AA, Gandomi, Amir H, 2021. Aquila optimizer: A novel meta-heuristic optimization algorithm. Computers & Industrial Engineering 157, 107250.