



Adaptive PCA-based feature drift detection using statistical measure

Supriya Agrahari¹ · Anil Kumar Singh¹

Received: 9 March 2022 / Revised: 9 March 2022 / Accepted: 19 July 2022 / Published online: 5 August 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

The plethora of existing methods in the streaming environment is sensitive to extensive and high-dimensional data. The distribution of these streaming data may change concerning time, known as concept drift. Several drift detectors are built to identify the drift near its occurrence point. Still, they lack proper attention to determine the feature relevance change over time, known as feature drift. Over time, the distribution change of the relevant features subset or the change in the relevant features subset itself may cause feature drift in the data stream. The paper proposes an adaptive principal component analysis based feature drift detection method (PCA-FDD) using the statistical measure to determine the feature drift. The proposed work presents a framework for identifying the most important features subset, feature drift, and incremental adaptation of the prediction model. The proposed method finds the relevant features subset by utilizing the incremental PCA and detects feature drift by observing the change in the percentage similarities among the most important features subset with respect to time. It also helps to forecast the prediction error of the base learning model. The proposed method is compared with state-of-the-art methods using synthetic and real-time datasets. The evaluation results exhibit that the proposed work performs better than the existing compared methods in terms of classification accuracy.

Keywords Data stream · Principal component analysis (PCA) · Feature drift · Concept drift · Prediction model

1 Introduction

With the advancement of technologies, many information society fields generate vast amounts of streaming data such as network access logs, weather forecasting data, medical data, traffic monitoring data, etc. The data stream is a set of data observations that arrives sequentially instance by instance. The traditional machine learning methods are based on the assumption of stationary data distribution. It means that the data is collected before the learning process. Whereas the streaming environment contains the non-stationary distribution of data. Traditional machine learning methods fail to handle continuously generating data. Thus, various methods have been developed to solve data stream problems. The change in the distribution of data instance

happens due to the dynamic nature of data over a period of time, known as concept drift [1].

There are different types of concept drifts like sudden or abrupt, incremental, gradual, recurring, and blip occurred in streaming data. These drifts are based on the speed of change in the data distribution concerning time. The concept drift negatively affects streaming data analysis and forecasting. In several drift detection methods, the detectors and learning model run simultaneously [2]. The detectors detect the distribution change in the incoming data instances, and the outcomes (or target values) of incoming data instances are predicted by the learning (or prediction) model. The learning model executes independent of the drift detection method, and its benefit is that it gives information about the dynamics of the generated data [3]. Due to the drift, it is seen that the decision boundary changes between the target (or class) values. Thus, the drift detection problem becomes more challenging because the training of the current prediction model is based on the old decision boundary. In this case, it wrongly classifies data instances (or examples) in the old class, whereas they generate from the new class. Several existing methods are based on monitoring the prediction error of the base

✉ Supriya Agrahari
supriyagrahari@gmail.com; supriyaagrahari@mnnit.ac.in

Anil Kumar Singh
ak@mnnit.ac.in

¹ MNNIT Allahabad, Prayagraj, India

learning model. Whereas the monitoring of data stream features may provide a better understanding of the evolution of concept change rather than finding the prediction error of the base learning model [4].

The existing literature has not been extensively addressed a specific kind of concept drift, i.e., feature drift, which signifies the possible changes in the relevance of data stream's features concerning time [5]. The feature selection has been substantially studied from a conventional mining perspective [6–8]. Still, it is challenging problem in the data stream domain because the concept drift and the varying length of the stream make impossible to apply the classical feature selection methods in the learning procedure [9]. The high-dimensional attribute vectors make a model computationally complex. In the streaming environment, the selection of appropriate data features becomes even more complicated.

Principal component analysis (PCA) has been successfully applied in data stream mining for dimensionality reduction. The limitation of PCA-based observation is that it is time-invariant. Whereas the streaming environment is time-varying. The time-varying characteristics of real-time data include (i) the changes in the mean, median, kurtosis, standard deviation, variance, etc., (ii) the changes in the correlation structure among data instances. If a time-invariant PCA model is used to monitor the data samples with the characteristics as mentioned earlier of real-time data, the false alarms are increased, and it significantly compromises the reliability of the pattern recognition in the streaming environment [10].

The paper uses an incremental principal component analysis (PCA) based method, which is time-variant, for feature drift detection. The principle components (PCs) extract the maximum percentage of explained variance as a cut-off point. PCs help to find the most important features subset and their respective distributions. The feature selection is necessary to avoid the irrelevant features because they are problematic as they require an extra and unnecessary computational cost for processing and storage. In addition to this, it makes classifiers more susceptible to overfit [11]. The feature drift is observed by calculating the significant change in percentage similarity of the mean and standard deviation of the most important features subset between two timestamps. In this way, the obtained features have the most impact on data distribution. It reduces the dimensionality of data to provide a way to detect the feature drift efficiently.

The feature drift may occur either due to the change in the distribution of the most important features subset, whether it is the same between two timestamps, or the change in the most important features subset itself concerning time. In this proposed work, three causes are taken into consideration for feature drift: (a) the distribution of the most important

features subset changes over time. (b) The relevance of the most important features subset itself changes concerning time. (c) The partial most important features subset may be similar, but their distribution changes with time. Thus, PCA-FDD determines the reason behind the drifts and detects the feature drift in the streaming environment. The main contributions of the paper are:

- We develop PCA-FDD, a feature drift detection method, which performs incremental drift detection and adaptation in a streaming environment.
- We develop a time-variant incremental principle component analysis to extract principal components (PCs) iteratively. PCA reduces the dimensionality of window data. In addition to this, we observe iterative identification of the number of Principal Components (PCs). It shows that PCs may change concerning time, and PCs help to extract the most important features subset from incoming data samples in the streaming environment.
- We experimentally evaluate the proposed and existing method using various synthetic and real-time datasets. The results show that PCA-FDD performs better than different existing methods in terms of the classification accuracy of the learning model.
- We utilize the Friedman test with Nemenyi-post-hoc analysis to verify the statistical significance of the performance of PCA-FDD and the compared methods using HT classifier. It shows that PCA-FDD is significantly better than the HDDDM methods.
- The experiment shows that PCA-FDD is a top-ranked method compared to the existing methods.

The remaining paper organizes as follows: preliminaries define in Sect. 2. Research-related work is discussed in Sect. 3. Section 4 presents the illustration of the proposed method. Experimental analysis and experimental environment are illustrated in Sect. 5. Section 6 discusses the result evaluation and statistical comparison of methods. Finally, the conclusion is presented in the last section.

2 Preliminaries

Concept drift Concept drift is a common attribute of data streams that occurs as a result of changes in the underlying contexts [18]. Concept refers to the joint probability distribution (P) of variables X and Y at timestamp t . A change in the joint probability distribution with respect to timestamp from t_l to t_m is considered as concept drift (see Eq. 1). Here, t_l and t_m represent two timestamps.

$$P_{t_l}(X, Y) \neq P_{t_m}(X, Y) \quad (1)$$

Feature drift Feature drift occurs when the relevance of features subset changes in the underlying distribution of

data concerning time. Suppose, the feature space at timestamp t is F_s and the relevant selected features subset is F_{s^*} , where $F_{s^*} \subseteq F_s$. A feature drift occurs (see Eq. 2) when the relevance of attributes of F_{s^*} changes between two timestamps t_l and t_m , i.e.,

$$(F_{s^*})_{t_l} \neq (F_{s^*})_{t_m} \quad (2)$$

Here, we consider a scenario of feature drift in streaming environment (see Fig. 1) and discuss three prospects as a cause of feature drift.

Figure 1a illustrates the set of features within a specific domain, and the feature space (F_s) contains the features like F_1, F_2, \dots, F_n . The particular prediction of a pattern change in the data stream will be based on F_s . Still, the relevant features subset (F_{s^*}) at timestamps t_l and t_m is subjected to change over time.

- When the distribution of the most important features subset is changed with time: Figure 1b shows the relevant features subset F_{s^*} (or $Fset$) at different timestamps t_l and t_m . The $Fset$ at different timestamps contains features $F_8, F_{21}, F_{40}, F_{43}$. Although the most important features subset is the same between timestamps but their distributions differ concerning time. This condition arises when data samples underline distributions changes over time due to internal/external factors.
- When the change in the most important features subset itself occurs concerning time: Figure 1c depicts $Fset = \{F_1, F_9, F_{21}, F_{40}\}$ at timestamp t_l and $Fset = \{F_8, F_{16}, F_{32}, F_{47}\}$ at timestamp t_m . It shows that the relevance of important features subset are changed, because the features may be influenced by various internal/external factors over time and the relevant features subset ceases to important in the future.
- When the partial set of most important features itself is changed, and the distribution of similar features is also changed over time: Figure 1d exhibits $Fset = \{F_{19}, F_{24}, F_{28}, F_{46}\}$ at timestamp t_l and $Fset = \{F_{19}, F_{24}, F_{37}, F_{51}\}$ at timestamp t_m . It shows that the partial features are the same and the remaining features are different. In this case, we identify the change in the distribution of similar features and if they are different, the feature drift condition arises in the data samples.

As a result, the method shows that the subset of the most important features may be relevant or non-relevant to the learning problem. The change in relevant important features subset impacts modification of the decision boundary. Thus, the learning model detects the feature drift and adapts accordingly [12]. So, instead of concerning distribution change only, these changes need to be incorporated to find the exact cause of feature drift in the streaming environment.

3 Research related work

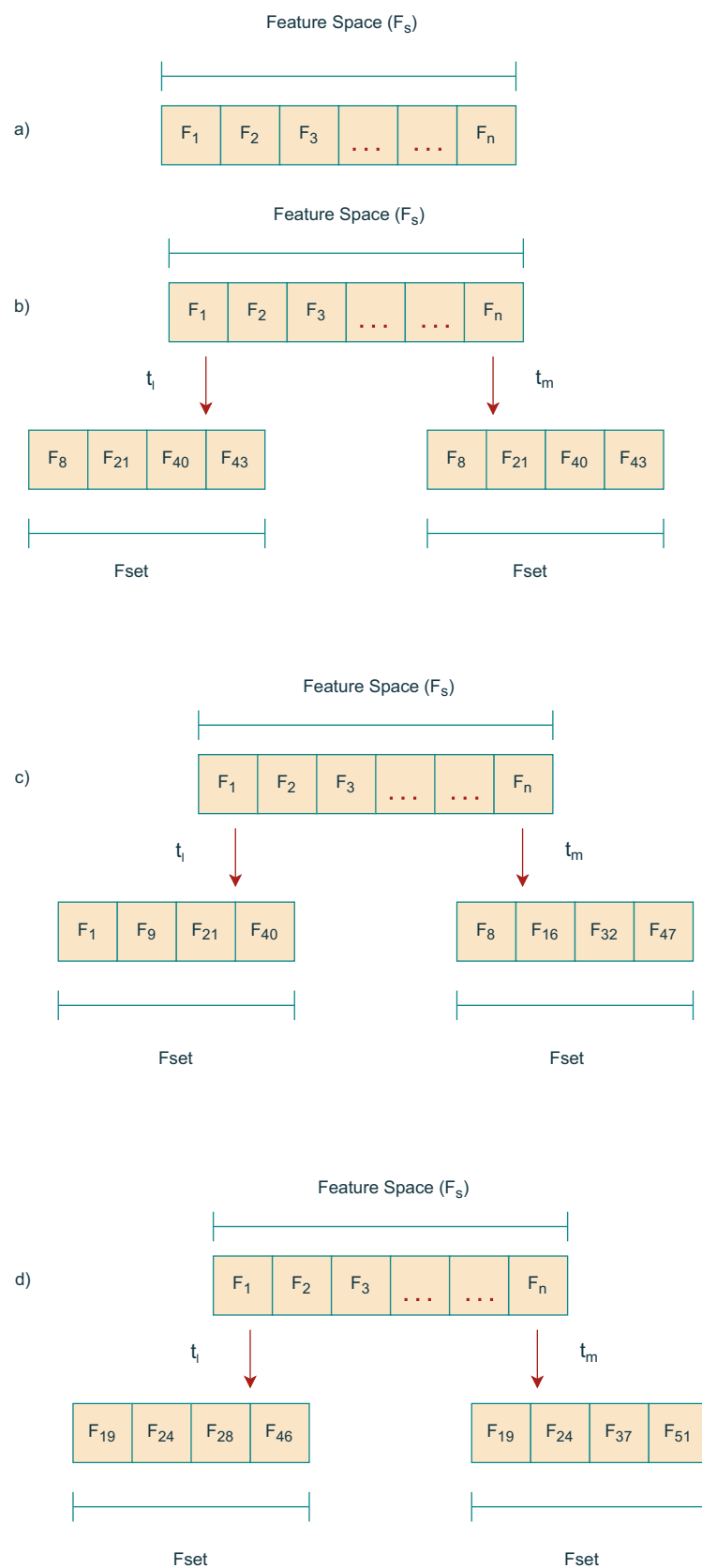
Most of the existing methods deal with characteristics like the infinite length of stream and concept drift in the streaming environment. However, they have not much focused on the feature drift problem. There are few existing works that perform feature selection during learning of drift detection in the data stream [4]. In this section, we discuss the existing methods that are based on feature relevance.

Feature Extraction for Explicit Concept Drift Detection (FEDD) [4] is an online explicit drift detection method. It identifies the drifts in the data stream by considering the features of data instances. The method contains a reference feature vector of the known context (or concept) and observes the evolution of this feature vector to identify the occurrence of drifts. FEDD considers two distance measures, the cosine distance and the Pearson correlation distance, used to evaluate the feature vector's dissimilarities. EDFs [13] is a novel unsupervised Ensemble Drift detector that acknowledges local changes in Feature Subspaces. It realizes the changes in the feature subspaces using incremental Kolmogorov–Smirnov tests. It reconstructs its subspaces after the detection of every drift. This way allows the capturing of new properties of the feature space. It limits the number of false positives produced by the sensitive univariate detections. In addition to this, EDFs encapsulates the several detectors in ensembles specialized in various feature subspaces to avoid complex multivariate computations.

Heterogeneous Ensemble with Feature drift for Data Streams (HEFT-Stream) [14] is a feature drift detection method. It assumes that the most discriminative features subset can be evaluated by filtering the disjoint chunks of examples. The major limitation of the method is how to find out the size of the windows. The size of the window directly influences the learning process. The procedure suggests that the small windows quickly recognize the possible changes in the chosen features subset. However, the method may detect a false change if the data stream contains noisy data. Whereas the large size of windows processes the huge amount of data, still it fails to detect a quick change in features relevances.

Another feature drift detection method is Landmark-based Feature Drift Detector (LFDD) [12]. It splits the data stream into chunks. LFDD finds the most discriminative feature subsets and trains the learning model. It presents a naive discriminative subset selection technique and does not consider the redundant features. Feature drift occurs when the most discriminative features subset of a data chunk varies from the features subset of the previous data chunk. Hellinger Distance Drift Detection Methodology

Fig. 1 Demonstration of various causes of feature drift



(HDDDM) [15] is an unlabeled feature tracking-based method. It performs the computation to check the change in the feature space. Two data distributions of data samples are analyzed by calculating the average Hellinger distance. The drift is signaled if the distance is greater than the defined threshold.

Margin Density Drift Detection (MD3) [16] is an incremental drift detection method. It finds the uncertainty or margin in the classifier's region by tracking the number of samples. It uses the metrics to detect a change in the data stream concept. The method is distribution and application-independent. MD3 using the SVM model (MD3-SVM) uses the support vector machine to find the margin. The model uses the linear kernel SVM with hinge loss. It tracks the number of data samples in the margin of a classifier to compute the margin density. Another model is MD3-RS which uses the random subspace (RS) drift detection technique of MD3. The ensemble comprises twenty decision trees. Each decision tree contains fifty percent of features randomly picked from the feature space. The number of data samples assesses the margin density in the ensemble's regions of high disagreement (or high uncertainty). The threshold is chosen as 0.5 for critical uncertainty. The data samples with confidence less than 0.5 are recognized within the margin.

Ding et al. [17] proposes an online entropy-based time-domain feature extraction method (ETFE) for concept drift detection. It performs drift detection using different steps: (a) the empirical mode decomposition based on extrema symmetric extension is considered to decompose the streaming data. Because of this, the different time scales features are adaptively extracted. (b) the entropy calculation is performed. The timestamp features are coarse-grained to quantify the structure and complexity of the data stream. (c) the statistical process control method based on generalized likelihood ratio is utilized to analyze the entropy change. It can efficiently find the mean and amplitude of the data stream. The method provides better robustness for drift detection.

Barddal et al. [5] develops DynamIc SymmetriCal Uncertainty Selection for Streams (DISCUSS), a merit-guided and classifier independent dynamic feature selection method. The computation of symmetrical uncertainty scores helps identify the redundant and irrelevant features. In addition to this, the single merit-guided sequential feature selection strategy is also proposed in this work. DISCUSS calculates the sum of relevant features as well as the sum of redundant features and sets allowable limits for it. The feature score is calculated concerning the class. Thus, it is not based on measuring the accuracy of the learning model.

An unsupervised drift detection method is Discriminative Drift Detector (D3) [18]. It uses a discriminative

classifier which is used with any online method without a built-in drift detection method. D3 arranges data instances in a sliding window to detect concept drift. The changes in the feature space monitor the concept drift.

One-class drift detector (OCDD) [19] uses a one-class learner with a sliding window to detect the concept drift. It is an unsupervised concept drift detection method. One-class classifier training distinguishes between the new data sample and the previous sample. It is a sliding window-based approach in which the drift is identified based on the percentage of the outliers present in the sliding window.

4 Proposed work

In the proposed work, principal component analysis (PCA) and the statistical analysis-based method introduce to identify the feature drifts in the data stream. The PCA performs the relevant feature extraction. Further, the percentage similarity of the mean and standard deviation of two data windows of relevant features is assessed to learn the associations among distributed data samples. In this method, PCA is used for dimensionality reduction before identifying the distribution change between data samples. It preserves n principal components (PCs) with most significant variance and discards other PCs to keep minimal redundancy and maximum information. It provides a better classification of data samples and identifies the patterns of incoming data examples. The advantage of PCA is that the sizeable dimensional data is reduced into small dimensional data without eliminating the important information. The pattern recognition process becomes more accurate and efficient.

The proposed method PCA-FDD uses PCA to find a subset of Principal Components (or principal directions) in a set of window data. PCA extracts the features by creating a smaller alternative set of variables. In this method, the window data is anticipated into the minor data set. The data examples of the dataset are cast into the Principal Component Spaces. The maximum percentage of explained variance extracts PCs as a cut-off point. The obtained PCs help to find the most important features subset. Generally, the comparison is performed by calculating the distance between these features subsets to detect the drift. The proposed work calculates the percentage similarity in the distribution of the most important features subset to find the feature drift. Here, pattern recognition is performed to determine the degree of similarity and inequality of the features subsets in the stream. The paper uses PCA with the similarity measures to achieve better pattern recognition results instead of using the similarity measure only.

In the proposed method, the detector finds the change in the pattern of data examples, and the prediction of data

examples is performed by the learning (or classification) model. The learning model includes the strategies to keep track of the most important subset of features in the feature selection method (i.e., PCA). The benefits of an accurate feature selection method are that the learner can process the examples faster using less memory space and present higher distinctness. These benefits are obtained due to the reduction in dimensionality, which needs fewer resources and retains only meaningful features for the learner training. The Pseudocode of PCA-FDD is illustrated in Algorithm 1, Algorithm 2, and Algorithm 3. Further, the workflow diagram of adaptive Principal Component Analysis based Feature Drift Detection method (PCA-FDD) is demonstrated in Fig. 2.

4.1 Relevant feature extraction

In the streaming environment, the feature space is not fixed concerning time. The features subset selection technique

gets full. Algorithm 2 performs feature selection and feature drift detection in the streaming environment. The proposed method utilizes the Principal Component Analysis (PCA) to extract the most important features subset. The PCA finds the subspace whose basis vectors correlate with the maximum data variance. It performs linear transformation to map the original D dimensional feature space onto a D' dimensional feature subspace. Here, D dimensional feature space depicts the number of dimensions (or features) present in a particular data sample, and D' dimensional feature subspace denotes the selected dimensions (or features) from the data sample after applying the feature selection method. In such condition, dimensional feature subspace (D') \leq dimensional feature space (D). In this way, PCA is used to extract the new feature subspace that better depicts the content of the data stream for the subsequent forecasting task.

Algorithm 1: Window Creation

Input: Incoming Instance: X , Current Window: W_c .

```

1 for each  $X$  do
2   if  $W_c == FULL$  then
3      $\lfloor$  FeatureSelection( $W_c$ )
4   else
5      $\lfloor$  Add instnace in  $W_c$ 

```

can be different for incoming data samples, and it creates other feature spaces of the classification model. So, there is a need to incorporate such changes during the classification of data examples. In this paper, the incremental PCA is used for dimensionality reduction of each data sample of the data stream. The aim is to retain the most important features that highly correlate with target class values and build a learning model with relevant features. Also, the minor features subset reduces the search space.

In Algorithm 1, the window is created by storing the incoming data instances (or data examples). The data examples are stored in the current window (W_c) until it gets full. The feature selection procedure initiates whenever W_c

The proposed method uses PCA for feature extraction before feature drift detection. The principal components (PCs) with the highest variance should preserve because the extracted features are more likely to be affected by the drift. In order to perform feature selection, all the data examples of a current window are taken into account, and PCA computes a vector that contains the most significant variance associated with it. With the use of PCA, we determine the number of components. The estimated number of components is required to describe the data sample. In this way, the dimensionality is reduced without much data loss. The method identifies the most impactful features that contribute maximum to the obtained components. In this way, we get an index of each of the most important features and their names.

Algorithm 2: Feature Selection and Feature Drift Detection

Input: Incoming Instance: X , Current Window: W_c , Percentage Similarity in Mean (μ): μ_m , Percentage Similarity in Standard Deviation (σ): σ_m , Temporary Variable: l , i , Threshold: θ , Similarity ratio: s_r , Initial window's $Fset$ Mean (μ): μ_{f_i} , Initial window's $Fset$ Standard Deviation (σ): σ_{f_i} , Current window's $Fset$ Mean (μ): μ_{f_c} , Current window's $Fset$ Standard Deviation (σ): σ_{f_c} .

Output: Most Importance Feature Subset: $Fset$, Feature Drift.

```

1 Function FeatureSelection( $W_c$ ):
2   Utilize Principle Component Analysis on  $W_c$  to extract num_components (PCs).
3   for  $i \leftarrow 1$  to num_components do
4      $Fset \leftarrow$  Extract most important features subset
5      $Fset_{norm} = \frac{Fset - Fset_{min}}{Fset_{max} - Fset_{min}}$  // Normalization of most important
        features subset
6     if  $l == 0$  then
7        $\mu_{f_i} \leftarrow \mu(Fset_{norm})$ 
8        $\sigma_{f_i} \leftarrow \sigma(Fset_{norm})$ 
9        $l = l + 1$ 
10    return False
11  else
12     $\mu_{f_c} \leftarrow \mu(Fset_{norm})$ 
13     $\sigma_{f_c} \leftarrow \sigma(Fset_{norm})$ 
14  Find  $Fset$  similarity between two window data instances and stored similarity
    ratio of  $Fset$  in  $s_r$ .
15  if  $s_r == 1$  then
16    // Calculate Percentage similarity among most important features
        subset
17     $\mu_m = \frac{len(set(\mu_{f_i}) \& set(\mu_{f_c}))}{len(set(\mu_{f_i}) \cup set(\mu_{f_c}))}$ 
18     $\sigma_m = \frac{len(set(\sigma_{f_i}) \& set(\sigma_{f_c}))}{len(set(\sigma_{f_i}) \cup set(\sigma_{f_c}))}$ 
19    // The distribution of the most important features subset is
        changed with time
20    if  $\mu_m \geq \theta \mid \sigma_m \geq \theta$  then
21      return False
22    else
23      Reset  $l$ 
24      return True
25    // The change in the most important features subset itself occurs
        concerning time
26  else if  $s_r == 0$  then
27    Reset  $l$ 
28    return True
29    // The partial set of most important features itself is changed and
        the distribution of similar features is changed over time
30  else
31    Find similar features from  $Fset$  and find  $\mu_m$  and  $\sigma_m$  of identical features.
32    if  $\mu_m \geq \theta \mid \sigma_m \geq \theta$  then
33      return False
34    else
35      Reset  $l$ 
36      The current window's  $Fset$  is used as the initial window for further
        computation with the new current window's  $Fset$ .
37      return True

```

4.2 Feature drift detection

The data stream is a continuous flow of data instances. It is required to run a model iteratively to analyze the behavior change of data concerning time. Thus, the proposed method utilizes PCA iteratively. The current window is

filled with incoming data instances each time, and it is compared with the initial window for feature drift detection. The proposed method considers the following situations as a cause of feature drift in the data stream:

(a) When the distribution of the most important features subset is changed with time: the PCA-FDD method

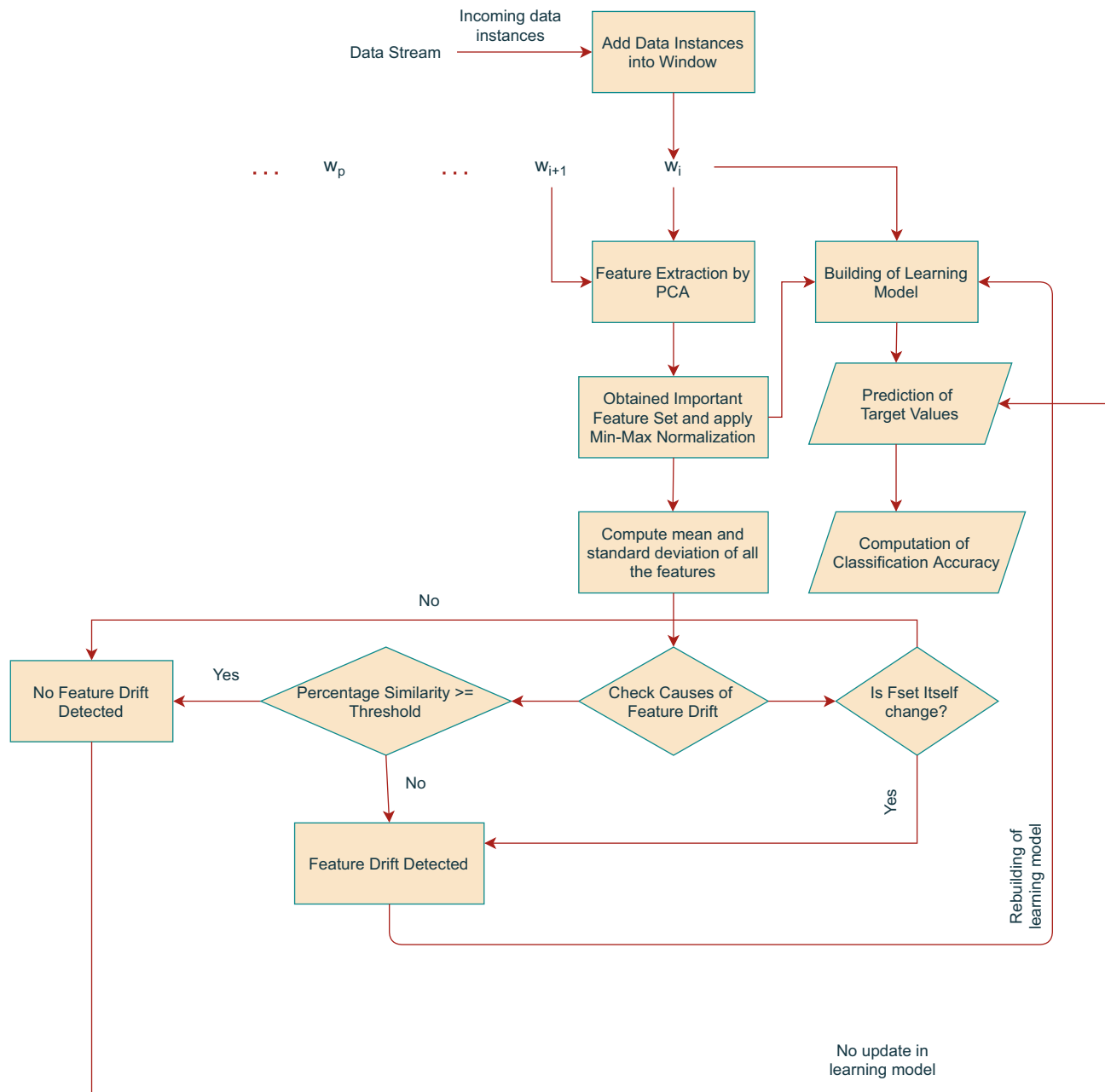


Fig. 2 Workflow diagram of adaptive principal component analysis based feature drift detection method (PCA-FDD)

compares the change in percentage similarity using mean and standard deviation between the most important features subset stored in initial and current windows to observe the distribution change of the most important features subset concerning time. The method normalizes the most important features subset using the min-max normalization technique (see Eq. 3). The scaling procedure is used to shift and rescale the data values in the normalization process. The obtained outcome is lied in the range $[0,1]$. The normalized important features subset demonstrates that each feature falls within the same range. It helps to ensure

that the features with enormous domains do not influence those with smaller domains.

Normalization is used when the data distribution does not observe the Gaussian distribution. It denotes that the method does not assume any underlying distribution of data instances. By applying the normalization in the most important features subset ($Fset$), we obtain a minor standard deviation and suppress the effect of outliers.

$$Fset_{norm} = \frac{Fset - Fset_{min}}{Fset_{max} - Fset_{min}} \quad (3)$$

Here, $Fset_{norm}$, $Fset_{min}$ and $Fset_{max}$ are the most important

features subset after normalization, minimum and maximum values of a feature, respectively. The mean (μ) and standard deviation (σ) of most important features subset ($Fset$) of initial and current window is stored in μ_{f_i} , σ_{f_i} , μ_{f_c} , and σ_{f_c} , respectively. In addition to this, the percentage similarities among data windows are observed to find the change between the feature subset's data distributions. The mean of window data examples shows the central tendency of data. Whereas the standard deviation of data shows the spread of data distribution. The percentage similarity of mean (μ_m) (see Eq. 4) and standard deviation (σ_m) (see Eq. 5) are computed by the following equations:

$$\mu_m = \frac{\text{len}(\text{set}(\mu_{f_i}) \cap \text{set}(\mu_{f_c}))}{\text{len}(\text{set}(\mu_{f_i}) \cup \text{set}(\mu_{f_c}))} \quad (4)$$

$$\sigma_m = \frac{\text{len}(\text{set}(\sigma_{f_i}) \cap \text{set}(\sigma_{f_c}))}{\text{len}(\text{set}(\sigma_{f_i}) \cup \text{set}(\sigma_{f_c}))} \quad (5)$$

The mean and standard deviation are used to show the statistics that reflect the data windows' data distribution in the pattern recognition process. The standard deviation of each feature in a diverse category changes distinctly. The obtained result from Eqs. 4 and 5 is used to determine the feature drift by comparing it with the threshold (θ). For analysis purposes, the greater or equal to 50% similarity percentage between two windows' most important features subset is considered as no feature drift condition; other-

most important features subset are different, and some are similar. In this case, the same procedure of feature drift detection is applied using Eqs. 4 and 5 and comparing the computed similarity measure to the threshold (θ) to find the significant change between data distribution of similar features.

In the proposed work, the feature drifts detect as per the above conditions.

Whenever the feature drifts encounter, the current window's most important features subset is used as a reference for further evaluation, and it becomes a new initial window. This process will continue until the data stream gets exhausted.

4.3 Analysis of classification accuracy

The data windows are created in real-time by collecting the incoming data instances for analysis purposes (see Algorithm 1). The learning (or classification) model predicts the target values of each data instance of the current window. The mean accuracy of the data window is calculated as a ratio of the number of correct predictions to the total number of predictions. The computation of mean accuracy defines in Eq. 6:

$$\text{MeanAccuracy}_i = \frac{\text{Number_of_correct_predictions}_i}{\text{Total_number_of_predictions}_i} \quad (6)$$

Algorithm 3: Classification Accuracy

Input: Data Stream: D_s , Incoming Instance: X , Current Window: W_c , Array: MeanAccuracy .

Output: Classification Accuracy.

1 Initialize $\text{MeanAccuracy} = \emptyset$.

2 **Function** ClassificationAccuracy(W_c):

3 **for each** $D_s \in W_c$ **do**

4 $\text{MeanAccuracy}_i = \frac{\text{Number_of_correct_predictions}_i}{\text{Total_number_of_predictions}_i}$

5 Append MeanAccuracy for each window

6 $\text{ClassificationAccuracy} = \frac{\sum_{i=1}^n \text{MeanAccuracy}_i}{\text{NumberofWindows}} * 100$

wise, the feature drift is detected.

(b) When the change in the most important features subset itself occurs concerning time: The most important features subset of initial and current window data examples is compared to observe the change in the most important features subset itself with time. If there are no similar features present in the features subset, it shows the occurrence of feature drift in the sequence of window data examples.

(c) When the partial set of most important features itself is changed, and the distribution of similar features is changed over time: In this condition, some features in the

The number of correct predictions is computed by comparing the predicted target values to the actual target values. It shows that the learning model predictions follow the current distribution of data examples. At the same time, the false predictions depict that the data distribution of data examples is changed. There is a possibility of relearning or rebuilding the prediction model as per the current distribution of data examples. In addition to this, the overall classification accuracy of the learning model is computed by Eq. 7:

$$\text{Classification Accuracy} = \frac{\sum_{i=1}^n \text{Mean Accuracy}_i}{\text{Number of Windows}} * 100 \quad (7)$$

Here, i denotes the i^{th} window, and n is the total number of the window created from the particular data stream. The computation of classification accuracy is performed to evaluate the overall classification model's performance and mean accuracy is computed to understand the incoming data pattern distribution behavior concerning the current classification (or learning) model.

In the proposed work, the feature drift detection is performed by considering the three causes discussed above in Sect. 4.2. The feature drift shows a change in the distribution of the most important features subset over time, or the most important features subset itself changes concerning time. After feature drifts detection, it is required to rebuild or relearn the classification (or prediction) model because it becomes obsolete for the current data examples. The distribution of the current data examples is required to be incorporated in the learning model to identify the data patterns efficiently. In this way, the proposed work builds an adaptive learning model.

5 Experimental analysis

This section presents experimental analyses. The experimentation of PCA-FDD is done on drift-induced datasets to understand better the proposed method's feature drift detection characteristics in a controlled environment. The synthetic and real-time datasets are used for feature drift detection. The following subsections demonstrate datasets descriptions and experimental environment.

5.1 Datasets

The data set is used for experimental purposes (see Table 1), which are taken from the UCI machine learning library. The data set is pre-processed numerical data. The seven data sets are evaluated. Out of seven, three are synthetic, and four are real-time data sets.

Synthetic datasets:

- Digits08 and Digits17 data set: A handwritten digit data set. Here, Digit08 contains 1499 instances and 16 attributes, whereas Digit17 contains 1557 instances and 16 attributes.
- Wine: The dataset contains a chemical analysis of wine that belongs to the same region of three different cultivars. It has 6497 instances and 12 attributes.

Real-time datasets:

- Forest Cover Type: The forest cover type data is a normalized dataset, and it covers 30*30 m cells in the area of US Forest Service (USFS), Region 2. It has fifty-four attributes, where forty-four attributes contain binary values, and ten attributes have numerical values. It exhibits various features like disappearances of vegetation, vegetation appearances, elevation, etc.
- Electricity: It is the electricity market data set of Australian New South Wales. The price of electricity depends on demand and supply. The target value is identified using price relative to moving average in 24 hours. It has 45312 instances and 08 attributes.
- Phishing: The dataset has information about phishing websites, and it is used to identify the phishing attack. It has 11055 instances and 46 attributes.
- Spam: It has a spam tagged message to identify whether the e-mail is legitimate or not. It has 6213 instances and 499 attributes.

5.2 Experimental environment

The proposed method PCA-FDD is built with the scikit-multiflow framework, a machine learning package for data stream in Python. For experimental purposes, the adaptive window size is taken into consideration. The window size shrinks when the feature drift occurs in the stream; otherwise, the window size is expanded. The Hoeffding Tree (HT) is used as a base classifier for evaluation purposes. PCA-FDD is compared with MD3 using SVM model (MD3-SVM), MD3 using random subspace model (MD3-RS), Hellinger distance drift detection methodology (HDDDM), discriminative drift detector (D3), one-class drift detector (OCDD), and static baseline model (NoChange) in terms of classification accuracy. The drift-induced data sets are used for evaluation purposes [16]. The drift in the dataset is induced by randomly collecting features subset and rotating their values for a particular class or target value.

Table 1 Datasets descriptions

Datasets	Number of instances	Number of attributes
Digits08	1499	16
Digits17	1557	16
Wine	6497	12
Forest cover type	218,515	54
Electricity	45,312	8
Phishing	11,055	46
Spam	6213	499

6 Result evaluation

This section presents experimental results and analysis with discussion and the statistical analysis of the proposed method with compared methods.

6.1 Experimental results and analyses

The experiment is performed using synthetic and real-time datasets. The proposed and compared methods MD3-SVM, MD3-RS, HDDDM, D3, OCDD, and NoChange are unlabelled drift detection methods. It means that the methods are based on distribution of input attributes. At the same time, the learning model's classification performance is achieved with labeled data. HDDDM relies on finding the changes in the raw feature values distribution. The classification accuracy determines the predictive performance of the learning model in the streaming environment. The number of drifts indicates the sensitivity to change in the data context.

The proposed and existing methods are compared in terms of the classification accuracy of the base learning model. The compared methods other than OCDD are feature extraction-based detection methods. All the existing methods are selected for comparison purposes due to fewer research-related articles on feature drift detection methods available and a good balance between old and new methods. The NoChange method assumes no drift present in the data stream regarding time. So, the model never gets updated and behaves like a static model for all the data instances.

The proposed method computes the classification accuracy as defined in Eq. 7. The comparison of proposed and existing methods in terms of classification accuracy using synthetic and real-time datasets are demonstrated in Tables 2 and 3, respectively. In the case of digits08, digits17 and the wine dataset, the proposed method outperforms as compared to existing methods. For the forest cover type dataset, PCA-FDD shows a significant increase in classification accuracy, whereas, with the electricity dataset, PCA-FDD and OCDD show similarity in performance. For the phishing dataset, HDDDM shows better performance. The spam dataset exhibits better performance with the MD3-RS method.

6.2 Statistical comparison of methods

For the verification of the statistical significance of the performance of PCA-FDD and the compared existing methods, Friedman test with Nemenyi-post-hoc analysis is performed [20]. The Friedman test defines the null hypothesis, illustrating that the equivalent methods share

the same rank. The paper evaluates the Friedman test with six methods using seven data sets. All the methods are arranged best to worst order in terms of classification accuracy, and the rank is assigned from 1 to k . The same rank is assigned when the performance of methods is the same. The average of their ranks assigns to them, if the classification accuracy of methods is similar. Here, k is the number of methods. The average rank of each method is demonstrated in Fig. 3. The figure shows that the proposed method PCA-FDD has the best rank and the NoChange method has the worst rank.

If the above null hypothesis is rejected, Nemenyi-post-hoc analysis can be conducted. It describes that the performance of the two methods is significantly different if the corresponding average ranks differ by at least critical distance (CD). It is computed in Eq. 8.

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} \quad (8)$$

Where q_α is the critical value and N is the number of datasets. The critical distance is obtained 3.09 by Eq. 8. Figure 4 shows the critical distance diagram of proposed and compared methods, and it is significantly better than HDDDM method. Figure 3 shows that PCA-FDD is the top-ranked method.

Table 2 Comparison of methods in terms of classification accuracy using synthetic dataset

Datasets	Methods	Classification accuracy	Drift detected
Digits08	PCA-FDD	95.6	1
	OCDD	93.7	2
	D3	93.6	1
	MD3-SVM	93.8	1
	MD3-RS	94.4	1
	HDDDM	93.5	2
	No Change	86.4	0
Digits17	PCA-FDD	96.4	1
	OCDD	92.3	1
	D3	92.6	1
	MD3-SVM	89.0	1
	MD3-RS	93.2	1
	HDDDM	88.7	2
	No Change	71.9	0
Wine	PCA-FDD	99.56	2
	OCDD	89.1	3
	D3	94.6	3
	MD3-SVM	96.9	1
	MD3-RS	94.9	1
	HDDDM	96.9	3
	No change	80.1	0

Bold signifies the highest performance of the method compared to existing methods in classification accuracy

Table 3 Comparison of methods in terms of classification accuracy using real-time dataset

Datasets	Methods	Classification accuracy	Drift detected
Forest cover type	PCA-FDD	85.2	23
	OCDD	83.1	26
	D3	84.9	25
	MD3-SVM	71.3	18
	MD3-RS	75.4	22
	HDDDM	74.9	25
	No change	62.2	0
Electricity	PCA-FDD	80.9	6
	OCDD	80.9	7
	D3	79.5	4
	MD3-SVM	66.9	2
	MD3-RS	67.7	2
	HDDDM	66.4	4
	No change	62.3	0
Phishing	PCA-FDD	90.9	4
	OCDD	90.8	3
	D3	89.4	4
	MD3-SVM	91.7	1
	MD3-RS	90.8	1
	HDDDM	92.8	4
	No change	86.9	0
Spam	PCA-FDD	87.8	2
	OCDD	86.6	4
	D3	87.0	4
	MD3-SVM	87.3	2
	MD3-RS	89.2	2
	HDDDM	80.7	2
	No change	57.5	0

Bold signifies the highest performance of the method compared to existing methods in classification accuracy

7 Conclusion

The paper proposes PCA-FDD, an adaptive principle components analysis based feature drift detection method. The area of feature drift is not extensively explored in the existing literature. The feature drift is a special kind of concept drift that occurs due to changes in the relevant features concerning time. In the proposed work, the drift detection is performed with the help of the most important features subset extracted from Principle Components iteratively for each data window instance. The paper considers the respective cause of feature drift. When the distribution of the most important features subset is changed with time or when the change is the most important features subset itself occurs concerning time. The PCA-FDD is compared with several state-of-art methods, namely MD3-SVM, MD3-RS, HDDDM, D3, OCDD, and NoChange, using three synthetic and four real-time datasets. The proposed

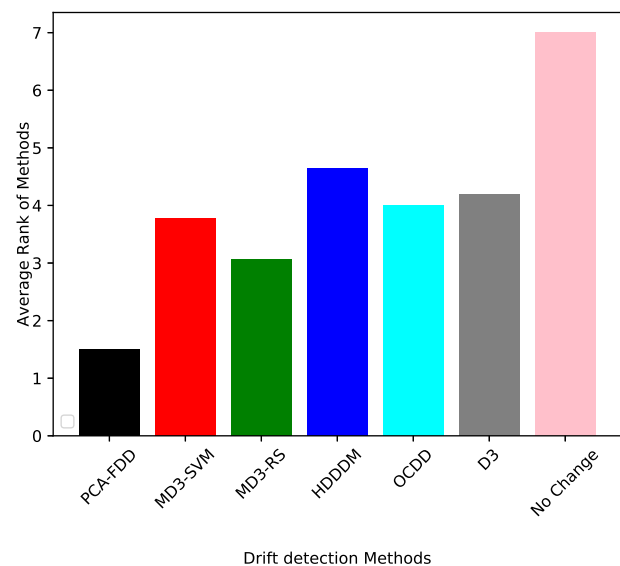
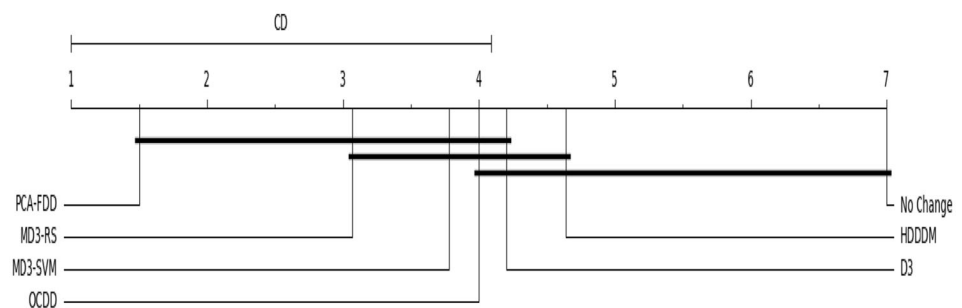
**Fig. 3** Average rank of methods

Fig. 4 Critical distance diagram based on classification accuracy of methods



method outperforms classification accuracy with Hoeffding Tree (HT) based classifier in the reported experiments. The Friedman test with Nemenyi-post-hoc analysis is performed to verify the statistical significance of the performance of PCA-FDD and the existing compared methods. The test suggests that PCA-FDD is significantly better than HDDDM method and a top-ranked method.

The future work is directed to study several future directions. The PCA-FDD can be applied in various domains to analyze the specific feature relevance changes with time. In addition to this, the implicit or explicit dynamic feature selection of data streams and the iteratively identifying feature drift with the different sizes of relevant feature subset are also challenging tasks. Further, the determination of varying classifier's behavior which deals with selected relevant features subset is also an open research direction. The noise filter methods can be helpful to deal with noisy attributes or classes. In addition, ensemble techniques can provide a promising opportunity to reduce the complexity and make a model more robust. In the windowing approach, the determination of the efficient size of the window is still a challenge.

Author contributions Not applicable

Funding Not applicable

Data availability Not applicable

Code availability Not applicable

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Ethical statement Not applicable

References

1. Agrahari, S., Singh, A.K.: Concept drift detection in data stream mining: a literature review. *J. King Saud Univ.* (2021)

2. Agrahari, S., Singh, A.K.: Disposition-based concept drift detection and adaptation in data stream. *Arab. J. Sci. Eng.* (2022). <https://doi.org/10.1007/s13369-022-06653-4>
3. Hammoodi, M., Stahl, F., Tennant, M.: Towards online concept drift detection with feature selection for data stream classification (2016)
4. Cavalcante, R.C., Minku, L.L., Oliveira, A.L.: Fedd: feature extraction for explicit concept drift detection in time series. In: 2016 International Joint Conference on Neural Networks (IJCNN), IEEE, pp. 740–747 (2016)
5. Barddal, J.P., Enembreck, F., Gomes, H.M., Bifet, A., Pfahringer, B.: Merit-guided dynamic feature selection filter for data streams. *Expert Syst. Appl.* **116**, 227–242 (2019)
6. Hammoodi, M.S., Stahl, F., Badii, A.: Real-time feature selection technique with concept drift detection using adaptive micro-clusters for data stream mining. *Knowl. Based Syst.* **161**, 205–239 (2018)
7. Zhou, P., Hu, X., Li, P., Wu, X.: Ofs-density: a novel online streaming feature selection method. *Pattern Recogn.* **86**, 48–61 (2019)
8. BenSaid, F., Alimi, A.M.: Online feature selection system for big data classification based on multi-objective automated negotiation. *Pattern Recogn.* **110**, 107629 (2021)
9. Turkov, P., Krasotkina, O., Mottl, V., Sychugov, A.: Feature selection for handling concept drift in the data stream classification. In: International conference on machine learning and data mining in pattern recognition. Springer, pp. 614–629 (2016)
10. Li, W., Yue, H.H., Valle-Cervantes, S., Qin, S.J.: Recursive PCA for adaptive process monitoring. *J. Process Control* **10**(5), 471–486 (2000)
11. Yu, L., Liu, H.: Feature selection for high-dimensional data: A fast correlation-based filter solution. In: Proceedings of the 20th international conference on machine learning (ICML-03), pp. 856–863 (2003)
12. Barddal, J.P., Gomes, H.M., Enembreck, F., Pfahringer, B.: A survey on feature drift adaptation: definition, benchmark, challenges and future directions. *J. Syst. Softw.* **127**, 278–294 (2017)
13. Korycki, L., Krawczyk, B.: Unsupervised drift detector ensembles for data stream mining. In: 2019 IEEE international conference on data science and advanced analytics (DSAA). IEEE, pp. 317–325 (2019)
14. Nguyen, H.-L., Woon, Y.-K., Ng, W.-K., Wan, L.: Heterogeneous ensemble for feature drifts in data streams. In: Pacific-Asia conference on knowledge discovery and data mining. Springer, pp. 1–12 (2012)
15. Ditzler, G., Polikar, R.: Hellinger distance based drift detection for nonstationary environments. In: IEEE symposium on computational intelligence in dynamic and uncertain environments (CIDUE). IEEE, pp. 41–48 (2011)
16. Sethi, T.S., Kantardzic, M.: On the reliable detection of concept drift from streaming unlabeled data. *Expert Syst. Appl.* **82**, 77–99 (2017)

17. Ding, F., Luo, C.: The entropy-based time domain feature extraction for online concept drift detection. *Entropy* **21**(12), 1187 (2019)
18. Gözüaık, Ö., Büyükakır, A., Bonab, H., Can, F.: Unsupervised concept drift detection with a discriminative classifier. In: *Proceedings of the 28th ACM international conference on information and knowledge management*, pp. 2365–2368 (2019)
19. Gözüaık, Ö., Can, F.: Concept learning using one-class classifiers for implicit drift detection in evolving data streams. *Artif. Intell. Rev.* **54**(5), 3725–3747 (2021)
20. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* **7**, 1–30 (2006)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.



Supriya Agrahari has done her Bachelor of Engineering (B.E.) in Information Technology in 2009 and Masters of Technology (M.Tech) in 2012 from National Institute of Technology Rourkela. She is currently doing her Ph.D. in the Department of Computer Science and Engineering from Motilal Nehru National Institute of Technology Allahabad. She is getting fellowship (under MHRD) in Ph.D. Ms. Supriya has her research interests and

publications in the areas of Data Stream Mining, Concept Drift and Pattern Recognition.



Anil Kumar Singh has done his Bachelor's in Science in 1990 and Masters in Computer Application (MCA) in 1994 from the University of Lucknow. He has also done his Masters in Engineering (Computer Sci. and Engg.) from University of Allahabad in 2001. He got his Ph.D. (Computer Science and Engineering) degree from Indian Institute of Technology Roorkee, India in 2012. He is currently working as Professor in the Department of Computer Science and Engineering since. Dr. Singh has his research interests and publications in the areas of Data Mining, Semantic Web, Information Retrieval, Machine Learning and Big Data Analytics. He was nominated & awarded Common wealth Scholarship, 2004 CANADA by MHRD, INDIA. He has also received AICTE fellowship (under QIP) during PhD.