Computers, Materials & Continua
DOI: 10.32604/cmc.2023.035221
Article

Tech Science Press

Drift Detection Method Using Distance Measures and Windowing Schemes for Sentiment Classification

Idris Rabiu[1,3,*], Naomie Salim[2], Maged Nasser[1,4], Aminu Da'u[1], Taiseer Abdalla Elfadil Eisa[5] and Mhassen Elnour Elneel Dalam[6]

[1]School of Computing, Univerti Teknologi Malaysia, Johor, 81310, Malaysia
[2]UTM Big Data Centre, Ibnu Sina Institute for Scientific and Industrial Research, Universiti Teknologi Malaysia, Johor, 81310, Malaysia
[3]Ibrahim Badamasi Babangida University, Lapai, PMB 11, Niger Steate, Nigeria
[4]UNITAR Graduate School, UNITAR International University, Tierra Crest, Jln SS6//3, Petaling Jaya, 47301, Selangor, Malaysia
[5]Department of Information Systems-Girls Section, King Khalid University, Mahayil, 62529, Saudi Arabia
[6]Department of Mathematics-Girls Section, King Khalid University, Mahayil, 62529, Saudi Arabia
*Corresponding Author: Idris Rabiu. Email: idrisrabiu43@gmail.com
Received: 12 August 2022; Accepted: 12 November 2022

Abstract: Textual data streams have been extensively used in practical applications where consumers of online products have expressed their views regarding online products. Due to changes in data distribution, commonly referred to as concept drift, mining this data stream is a challenging problem for researchers. The majority of the existing drift detection techniques are based on classification errors, which have higher probabilities of false-positive or missed detections. To improve classification accuracy, there is a need to develop more intuitive detection techniques that can identify a great number of drifts in the data streams. This paper presents an adaptive unsupervised learning technique, an ensemble classifier based on drift detection for opinion mining and sentiment classification. To improve classification performance, this approach uses four different dissimilarity measures to determine the degree of concept drifts in the data stream. Whenever a drift is detected, the proposed method builds and adds a new classifier to the ensemble. To add a new classifier, the total number of classifiers in the ensemble is first checked if the limit is exceeded before the classifier with the least weight is removed from the ensemble. To this end, a weighting mechanism is used to calculate the weight of each classifier, which decides the contribution of each classifier in the final classification results. Several experiments were conducted on real-world datasets and the results were evaluated on the false positive rate, miss detection rate, and accuracy measures. The proposed method is also compared with the state-of-the-art methods, which include DDM, EDDM, and PageHinkley with support vector machine (SVM) and Naïve Bayes classifiers that are frequently used in concept drift detection studies. In all cases, the results show the efficiency of our proposed method.

Keywords: Data streams; sentiment analysis; concept drift; ensemble classification; adaptive window

1 Introduction

Data streams are sequences of samples that were ordered and created continuously in real-time [1]. The example of data streams include recommender systems, financial time series, network traffic data, and other sensor data. In addition, one of the most popular stream models is the classification model for data streams, which identifies and classifies a set of categories to which an observation belongs [2]. Unlike traditional stationary settings, stream data differ significantly due to issues like an increase in data volume, read-only access, imbalanced scenarios, and concept drift issues [3].

In machine learning and data mining, the concept drift problem occurs when the relationships between input and output data change over time [4]. This problem has become an attractive research topic that concerns multidisciplinary domains such as data mining, machine learning, statistic decision theory, and ubiquitous knowledge discovery among others [5]. In these domains, authors have referred to concept drift issues by a variety of names, which include idea drift, dataset shift, covariate shift, and non-stationarity, among others [4]. Therefore, this paper employs the term concept drift to refer to shifts in user's opinion toward the online products in this paper. In general, concept drift can occur in different patterns such as abrupt drift, gradual drift, incremental, and recurring drifts based on the rate of changes [6].

In recent times, sentiment analysis has been one of the research areas that grows exponentially as a result of the increased availability of online user-generated reviews. Due to the unpredictability of review content and evolving user perception of items, concept drift concerns have become increasingly difficult in text stream analysis, particularly review-based sentiment categorization [7,8]. Users occasionally give their thoughts on a certain item based on its qualities which often change over time. For example, a person may have a favorable view of a phone gadget, but if a significant function changes (is added or removed), certain terms related to the new function might suddenly appear or disappear in the user's review, indicating a different view. The underlying data distribution changes as a result of these dynamic environment changes, resulting in poor classifier performance [9,10].

There are several categories of methods that were introduced to address the concept drift issues which are window-based approaches, weight-based approaches, and ensemble-based approaches [6]. Although the ensemble classifier is one of the most effective and widely used classification approaches, dealing with large and dynamic input streams requires a more complex approach. Furthermore, the ensemble strategy combines many basic methods, utilizing the benefits of their combined performances, to improve prediction ability beyond what any of the individual methods can achieve [11,12]. Recently, a number of studies have encouraged the ensemble method, with notable success [11–13]. In light of their effectiveness, the selected ensemble paradigms have been adapted for data stream mining [14–16]. In spite of that, the majority of these methods are limited to base classifiers to enhance the predictive performance of data streams, ignoring the concept of drift detection [17,18], which enable us to further explore and focus more on this aspect.

Instead of using a static ensemble of classifiers, this paper proposes a novel ensemble classifier that combines drift detectors to simultaneously determine and detect concept drifts while classifying user sentiments to improve accuracy. Therefore, this paper is expected to contribute based on the following aspects:

- A concept drift detection technique is used to provide timely responses toward detecting the concept drift of user sentiments. In addition, a two-window technique is used to detect concept drift.
- Different dissimilarity measures are investigated to address concept drift problems based on their performance in measuring the distribution between two consecutive windows, which are known as the reference window and current window. In addition, a novel framework for adaptive ensemble classification is developed, considering the application of concept drift to increase the ensemble classifier's predictive performance.
- Several experiments on real-world datasets were carried out, and the findings show that the suggested method outperforms the benchmark models in terms of accuracy.

The following are the remaining sections of the paper: Section 2 describes a literature review of existing studies. Next, Section 3 elaborates on the proposed method. Then, Section 4 evaluates the proposed model's performance through experiments. Finally, Section 5 discusses the conclusion of the paper.

2 Related Works

The main focus of this section is to review the relevant studies regarding sentiment analysis and concept drift detection approaches. Correspondingly, this paper aims to investigate how to employ textual data features for the detection of concept drift of user opinions, which are prominent in real-world online applications and becomes a major challenge to classification accuracy [19,20]. Concept drift detection approaches are typically used in conjunction with a base classifier, such as the NB and LibSVM (SVM) models to increase classification accuracy [20–23]. Stream classification models, in general, are designed to train classifiers on both historical and current instances in the stream in order to predict the label sets of incoming instances [7].

However, data instances arrived at a higher rate, and therefore require the classifiers to process them with stringent time and memory constraints [13]. Hence, assemble models for streaming contexts are being developed to increase classification accuracy to address these issues [24–26]. Existing ensemble techniques generally improve the problem of prediction accuracy by training individual classifiers based on different sets of data examples and combining them to predict incoming instances using predefined weighting algorithms [24,27]. Yet, such strategies were ineffective in dealing with concept drift.

Generally, two categories of concept drift detection are discussed: evolving-based and trigger-based learners [4]. With a dynamic learning approach, learners are periodically updated regardless of whether or not a change has taken place indicating that, evolving learners are unable to identify changes explicitly and, have limited control over how the new model should be reconstructed. In this category, there are two approaches: instance selection and instance weighting. Instance selection deals with the selection of instances that are most relevant to the current concept. The most frequent variation of the instance selection method is the time window approach, which moves a variable or constant-size window across newly arrived instances [10]. In contrast, instance weighting determines the weighting of instances using decay functions proportional to the instances' age and relevance to the existing concept [10].

The trigger-based learners are the second most common type of adaption approach. The trigger-based technique deals simultaneously with detection models that create signals suggesting the need to modify the current model. There are two categories of trigger-based models; the first group generates signals based on information related to the classifier's performance, whereas the second group uses dissimilarity measures to monitor changes in data distributions. For instance, Gama et al. [6] provide the models to detect concept drift by leveraging error rate-based stream data from the classifier as a change indicator. When the error rate exceeds the threshold, an indicator of concept drift is generated, indicating that the model needs to be updated. Similarly, Margaris et al. [28] utilized the interval rate between accurate and inaccurate predictions to detect concept shifts. This interval is sensitive to change detections, allowing the method to operate effectively with many types of drift, particularly slow change types as opposed to rapid change types. In addition, Liu et al. [29] investigated the differences in accuracy between recent occurrences and overall data to detect concept drift. However, if the sliding window is too small, this strategy generates higher false alarms, and if the sliding window is too huge, it will not function correctly in the presence of gradual drifts.

Another detection method uses batch mode to observe two different distributions. This method employs a fixed reference window to summarize historical data and a sliding detection window to cover the recent samples. Therefore, the null hypothesis H_0, which suggests the distributions are identical, whereas the alternative hypothesis H_1, which implies the occurrences of a change, is used to compare distributions throughout these two time periods using dissimilarity measures and statistical tests [30]. The following are the most commonly used measures: Hellinger distance (HD), Kullback–Leibler divergence (KD), Total variation distance (TVD), and the Kolmogorov–Smirnov statistic (KS distance) [30]. Unlike the aforementioned methods, this method detects concept drift independent of the classification error rate. Thus, our method makes use of these methodologies and an ensemble classifier, which is designed to simultaneously detect multiple forms of drifts. To sum up, our method differs significantly from previous single-classifier drift detection-based methods and ensemble classifiers without drift detection methods.

3 Proposed Drift Detection Method Based on Adaptive Windowing (DDAW)

This paper explores the proposed DDAW approach for concept drift detection. It is an unsupervised concept drift detection algorithm that sets a trigger as an indicator of concept drift. DDAW is built primarily to handle concept drift in user opinion and sentiment analysis based on textual reviews. It tries to improve the ensemble model's ability to deal with concept drift and maximize the model's precision. Fig. 1 expands on the principal components of drift detection methods.

3.1 Feature Extraction

The initial step of the DDAW algorithm, as represented in Fig. 1, is to extract the textual features from the text input using Natural Language Processing (NLP) techniques. Specifically, our approach starts by cleaning and preparing the text data by stripping it of all unwanted wanted characters such as HTML markup characters, stop words, punctuation marks, and other non-letter characters. This is accomplished using a regular expression (reex) library in python. After successfully preparing the review datasets, the next step is to tokenize the text documents into individual elements using the cleaned documents at their whitespace characters. For example, consider a review comment: "This movie is fantastic! I really like it". By applying the cleaning and tokenization processes, the above sentence is represented as ['movie', 'fantastic', 'really', 'like']. Then, the resulting document is applied to a Stanford POS Tagger, which reads each token and assigns part of a speech tag to each word, such as noun, verb, adjective, etc. [31]. In this paper, the first three tags are considered (noun, verb, and adjective), which are the most commonly used while reviewing sentiments of a sentence as they carry the most valuable information regarding reviewed items. Based on the above sentence, the POS tag is applied based on the noun, verb, and adjective tags as follows; [NN-'movie', JJ-'fantastic', JJ-'really', VB-'like']. Next, this paper uses a tf-idf to represent each term and form vectors. Generally, the vocabulary of the corpus is first built based on the pre-processed document and then generate the word count vector from each review based on the frequency of words present in the vocabulary. Based on the vocabulary generated, the word count vector is generated from each review sentence based on the frequency of words present in the vocabulary. Then, tf-idf is calculated as the product of the term frequency and the inverse document frequency using Eq. (1).

$$tf - idf\ (t, d) = tf\ (t, d) \times idf\ (t, d) \tag{1}$$

Here $tf - idf\ (t, d)$ is the term frequency, and $idf\ (t, d)$ is the inverse document frequency and can be calculated as in Eq. (2):

$$idf\ (t, d) = log\frac{n_d}{1 + df\ (d, t)} \tag{2}$$

The resulting $tf - idf\ (t, d)$ features are then used to classify the review data as positive or negative sentiments using the proposed ensemble model.

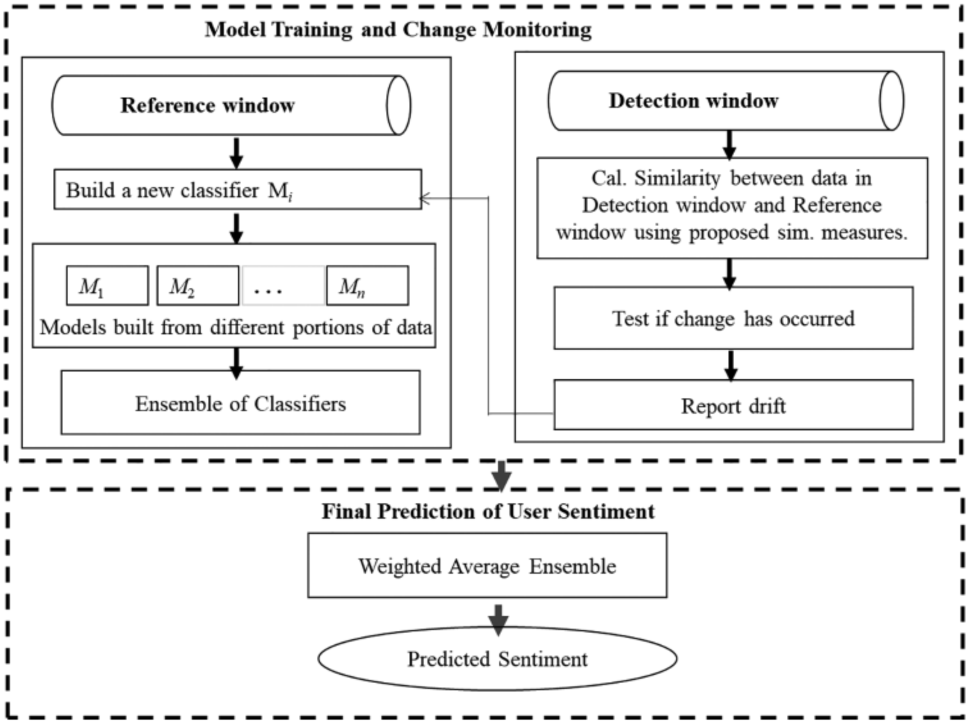
Model Training and Change Monitoring
Reference window
Build a new classifier M_i
M_1
M_2
...
M_n
Models built from different portions of data
Ensemble of Classifiers
Detection window
Cal. Similarity between data in Detection window and Reference window using proposed sim. measures.
Test if change has occurred
Report drift
Final Prediction of User Sentiment
Weighted Average Ensemble
Predicted Sentiment

Figure 1: The structure of the proposed ensemble learning models

3.2 Training Phase

The main objective is to train classifiers to predict the sentiment of each review in the stream. In this phase, the data stream was first generated using the extracted document term matrices obtained from Section 4.1 above by segmenting the data into a window with a fixed number of instances taken from equal time intervals. The first window is used as the training window, since it is used in the training process of the classifiers, and formulated in the learning process. As shown in Fig. 1, a number of component classifiers are built based on different segments of the input stream, and a pool of classifiers (each classifier representing one of the current concepts) is maintained to predict the class of incoming instances using a weighted average ensemble approach.

A drift detection module was utilized to follow changes in the data stream by monitoring the differences between two consecutive windows, one represents older instances while the other one reflects the recent occurrences. A new classifier is trained after an occurrence of a change, and a new concept is discovered and included in the pool. Meanwhile, the archived historical concepts are being assessed for potential reuse.

As new streams of data arrived and more concept drifts were identified, a new classifier was constructed using window W2, weighted, and added to the ensemble. Prior to adding a new classifier, it was assumed that new concepts were recovered. Therefore, the total number of classifiers in the ensemble is tracked to determine if it has exceeded the specified maximum limit K, and if it has, the classifier of the lowest weight is excluded from the ensemble.

3.3 Drift Detection-Based on Adaptive Window Model

As previously mentioned, most of the concept drift detection techniques employ classifiers or learning models. In addition, the majority of change detection techniques prioritize change detections based on classifiers' error rates but ignore changes in data distribution [21,32,33]. Contrary to other data streams, detecting the concept drifts in text streams is more challenging for several reasons. Firstly, a concept drift is mostly detected depending on the classifiers' performances. However, the text stream is characterized by sparsity and high dimensionality, which can cause the deviation of classification error rates to be high, and consequently makes the drift detection based on the error rates to be less accurate. Besides, concept drifts take different forms and not all changes can be directly and appropriately reflected in the error rates [20]. In addition, the changes in error rates may take gradual processes and the false predicted instances must be accumulated enough before a concept drift can be detected. To overcome these problems, this paper proposed a novel concept drift detection method by employing a two-window strategy to compare the data distribution across two consecutive windows that are considered the most prevalent method based on the literature review [34,35].

Where W_1 and W_2 stand for reference and current windows, respectively. As depicted in Fig. 2, the size of both windows is n. In most cases, the null hypothesis H_0 is selected against the alternative hypothesis H_1 in the problem of change detection in data streams, shown as follows:

$$\begin{cases} H_0 \; d\left(W_1, W_2\right) \leq \varepsilon \\ H_1 \; d\left(W_1, W_2\right) > \varepsilon \end{cases} \tag{3}$$

where $d\left(W_1, W_2\right)$ denotes a distance function that measures the dissimilarity of two-time windows and the parameter ε resents a distance-based threshold to determine if a change has occurred. If the measure of dissimilarity between two windows exceeds a certain threshold, then a change has occurred.

Four distance measures namely Hellinger distance (HD), Kullback–Leibler divergence: (KD), Total variation distance (TVD), and the Kolmogorov–Smirnov statistic, are investigated by comparing the current window with the reference window according to their effectiveness toward drift detection [30]. Each of these is described in detail as follows:

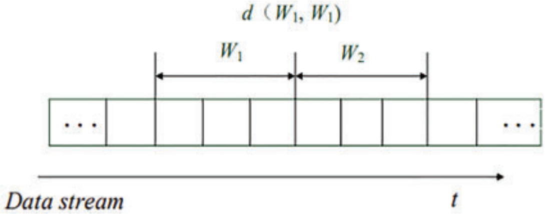

Figure 2: Two-window change detection model

3.3.1 Hellinger Distance

This metric was introduced by Ernst Hellinger in 1909, to measure the dissimilarity between two distributions. Let and be two discrete distributions. Hellinger distance between W_1 and W_2 is defined as:

$$d\left(W_1, W_2\right) = \frac{1}{\sqrt{2}}\sqrt{\sum\nolimits_{i=1}^{k}\left(\sqrt{W_{1i}}\ -\ W_{2i}\right)^2} \tag{4}$$

Unlike other measures, the Hellinger distance satisfied a triangle inequality. The inclusion of $\sqrt{2}$ in the definition of Hellinger distance is to confirm that the distance value is always between 0 and 1.

3.3.2 Kullback–Leibler Divergence

Kullback–Leibler (KL) divergence is known as an information-based measure of disparity among probability distributions [36]. If the data samples are continuous over the set, then, KL divergence between probability distributions is denoted by:

$$d\left(W_1, W_2\right) = -\int log_2\left(\frac{W_1\left(p\right)}{W_2\left(p\right)}dP\right) \tag{5}$$

For discrete distributions:

$$d\left(W_1, W_2\right) = \sum\nolimits_i W_{1i}log\left(\frac{W_{1i}}{W_{2i}}\right) \tag{6}$$

where $d\left(W_1, W_2\right)$ is the similarity measure between W_1 and W_2, and could either be continuous or discrete as the case may be.

3.3.3 Total Variation Distance

Total variation distance (TVD) is known as data distribution dissimilarity measure [30]. TVD is commonly referred to as "the statistical distance," which is mathematically is defined as follows:

$$d\left(W_1, W_2\right) = \frac{1}{2}\sum\nolimits_i \left|W_{1i} - W_{2i}\right| \tag{7}$$

where $d\left(W_1, W_2\right)$ is the similarity measure between W_1 and W_2, respectively.

3.3.4 Kolmogorov–Smirnov Statistic

The Kolmogorov–Smirnov statistic (KS distance) measures the distance between two probability distributions on a single real variable [37]. It can be used to calculate the distance between two samples or between a sample and a distribution defined as:

$$D_{n,n'} = \underbrace{Sup}_{p} \left| F_{1,n}(p) - F_{2,n'}(p) \right| \tag{8}$$

where $D_{n,n'}$ is the similarity between W_1 and W_2, $F_{1,n}(p)$ and $F_{2,n'}(p)$ are the empirical cumulative distribution function (ECDF) for observations W_1 and W_2.

The procedure for the proposed drift detection-based adaptive windowing for sentiment classification approach can be given in Algorithm 1 as follows:

Algorithm 1: DDAW algorithm for the sentiment classification

```
Input:
    $S$: stream data;
    $\varepsilon$: the threshold;
    $K$: the size of the ensemble classifier
Output:
    Drift: the number of concept drifts;
    Ensemble: the ensemble of classifiers;
Begin
  Ensemble ← ∅
  Drift ← ∅
  Read a window $W_i$ from stream $S$ and train classifier $C_i$ with $W_i$;
  While $S$ is not at the end do
      read the next chunk to $W_i$;
      train the base classifier $C_i$ from $W_i$;
      calculate the drift rate of $W_{i-1}$ and $W_i$;
      if $d(W_{i-1}, W_i) > \varepsilon$ then
            create a new classifier $C_i$;
              update the weight of all classifiers in the ensemble;
        if |Ensemble| $< K$, then Ensemble ← Ensemble ∪ $C_i$;
            else prune the worst classifier;
        else if the concept is recurring
            reuse the classifier in Ensemble;
              end if
          end if
        end if
     end
end
```

Our technique is based on the proposed Algorithm 1 and consists primarily of two steps: concept drift detection and classifier training. Similar to [24], our method initially detects concept drifts when a new instance W_2 is received, then trains a new classifier C_2 for W_2 once a change is detected. To add new classifier, the total number of classifiers in the ensemble is checked if the limit is exceeded, before the classifier with the least weight is removed from the ensemble. For each member of the classifier, the weights can be calculated based on their mean square error (MSE) estimates on the new data d_i and reference data d_r as shown in Eq. (9) [24].

$$Weight_i = \frac{1}{MSE_r + MSE_i + \alpha} \tag{9}$$

where $Weight_i$ is the weight of the classifier C_i, and α is added to avoid division by zero. However, when the concept remains unchanged, the old classifier is reapplied. In contrast to conventional models that begin by training the classifiers, obtaining the error rate, and then detecting drifts, this approach begins by detecting the drifts. Although the training of classifiers follows similar methods, the method for detecting drifts is distinct. Using the similarity measurements, Algorithm 1 will be repeated four times for various dissimilarity measures in our proposed DDAW technique, which will used each time to establish the optimal way of quantifying the drift magnitude.

4 Performance Evaluation

In this section, the analysis of the experimental results and discussions is presented based on three perspectives: Investigation into the Performance of Concept Drift Detection Methods, Investigation into the Classification Results with Various Drift Detection Methods, and Model Sensitivity to Ensemble Size. However, the explanation about datasets, evaluation metrics, and parameter settings is first provided before the experimental results and discussions section.

4.1 Dataset

The proposed algorithm is applied to the streaming text data in two real-world datasets comprises of the Amazon shopping dataset obtained from [38] and the 20-Newsgroup dataset crawled by [20]. At first, the pre-processing on each of the datasets is performed. There are 6,400 instances of Amazon shopping data representing four different product categories: books, electronics, DVDs, and kitchen. Each product has 1,600 occurrences. To create a stream, 200 instances were selected at random from a subcategory (such as books) and read them many times. Concept drift is triggered from adjacent chunks from different sub-categories, whereas there is no concept drift for chunks from a similar sub-category. Amazon provides a text stream with three concept drifts by randomly selecting data from several categories. In addition, 20-Newsgroups is a news dataset with 10,000 occurrences and 16 subcategories. To create concept drifts, 200 instances were randomly chosen from multiple subcategories. This text stream is composed of 45 chunks and 33 drifts.

4.2 Evaluation Metrics and Baselines

This paper analyze the performance of our proposed method using three evaluation metrics: false alarm (FA), missing rate (MR), and error rate (ER). All of these measurements will be utilized to explore the effect of incorporating distance measures into DDAW detection algorithms. In addition, the proposed method isimplemented using the MATLAB R2020a platform and Gephi graph visualization program to illustrate the DDAW graphical findings. The experiments are conducted using Windows Vista machines as well as Intel Xeon CPUs (E5420 @ 2.5 GHz) and 12 GB of RAM. NB and SVM are used as our main classifiers in our experimental setup.

To pre-process the review text, a Stanford CoreNLP [39] is adopted to conduct word segmentation, POS tagging, and stemming. For example, consider a review comment: "This movie is fantastic! I really like it". By applying the cleaning and segmentation processes, the above sentence is represented as ['movie', 'fantastic', 'really', 'like']. Then the resulting document is applied to a POS tagger, which reads each of the tokens and assigns parts of a speech tag to each word, such as noun, verb, adjective, etc. [39]. Since not every word in a sentence contains the user sentiment, therefore, POS tagger helps to filter out such words. After that, the stop words are removed based on the list gathered, and the words that appear less than five times are also removed to further speed up the optimization process. Afterward, the Term Frequency-Inverse Document (TF-IDF) is used to summarize the ensuing review data.

Furthermore, to show the performance of our proposed algorithm, four versions of our algorithm were created based on distance methods. Therefore, these versions were compared in terms of three evaluation metrics.

DDAW-HD: This method refers to the DDAW algorithm based on Hellinger distance (DDAW-HD), which uses Hellinger distance to quantify the magnitude of drift or shift. Especially in unsupervised learning, distance measurements between distributions are required when covariate drift or shift occurs.

DDAW-KL: This method refers to the DDAW algorithm based on Kullback–Leibler divergence (DDAW-KL), which employs Kullback–Leibler divergence to calculate the magnitude of drift or shift between two distributions.

DDAW-TVD: This method refers to the DDAW algorithm based on Total variation distance (DDAW-TVD), to calculate the magnitude of drift or shift between two distributions.

DDAW-KSD: This method refers to the DDAW algorithm based on Kolmogorov–Smirnov distance statistic (DDAW-KSD), which employs Levenshtein edit distance to calculate the magnitude of drift between distributions.

Particularly, we chose three state-of-the-art models as the benchmarks for drift detections in data streams. These include Drift Detection Method (DDM), Early Drift Detection Method (EDDM) [40], and PageHinkley [41], which are the frequently used methods for concept drift detections, and served as baseline methods for a number of studies in this field.

4.3 Parameter Settings

This paper explain the significance of the parameters in our experiments and optimize the parameters of our suggested algorithm for optimal performance. These include the ensemble size, the window size, and the threshold for drift detection. The first input of the proposed model is a data stream of textual reviews that have been pre-processed using natural language processing techniques, and a document term matrix has been constructed. Here, the stream S is configured to contain 6,400 instances with four product kinds for the Amazon dataset and 10,000 instances with 16 subcategories for the 20-Newsgroups dataset. In the DDAW algorithm, different sizes of ensemble classifiers were considered. To begin with, the ensemble size k was changed from 5 to 20 ($k = 5, 7, 10, 12, 15, 18, 20$) to determine how it can affect the performance of the algorithms. Then, the value of $k = 10$ is picked as the accuracy of DDAW grows as the ensemble's number of classifiers increases, but becomes independent of ensemble size after $k = 10$. The window size has a minor impact on the performance of our proposed algorithm. Thus, the window size is set to 200 data chunk size for comparison purposes. This paper uses a distance-based criterion to determine whether or not a change has occurred by comparing the average of two windows.

4.4 Experimental Results and Discussion

This section present the findings based on our proposed DDAW algorithm by employing four different distance measures and then compare our results with the baseline approaches using the two real-world datasets namely, Amazon shopping [38] and 20-Newsgroup [20].

4.4.1 *Investigation into the Performance of Concept Drift Detection Methods*

Table 1 summarizes the experimental findings for three baseline methods and our proposed method that employs four distinct dissimilarity measures, for a total of ten (10) different approaches in terms of false alarm (FA) and missing rate (MR). Then, Figs. 3 and 4 exhibit the performance of drift detection methods on the Amazon shopping and 20-Newsgroup datasets for more clarity. The DDAW-HD model is superior to the other DDAW versions and the three based lines on both datasets, as shown in Table 1.

Table 1: Analysis of drift detection performance based on false alarm and miss rate

Metric	DDM		EDDM		Pagehinkely		DDAW-HD	DDAW-KL	DDAW-TD	DDAW-KD
	NB	SVM	NB	SVM	NB	SVM				
Amazon shopping										
False-alarm	4	5	5	4	6	5	1	3	3	2
Missing-rate	4	4	3	3	4	4	1	2	1	1
20-Newsgroups										
False-alarm	9	10	8	9	8	7	3	4	5	4
Missing-rate	11	12	11	11	10	13	2	6	7	6

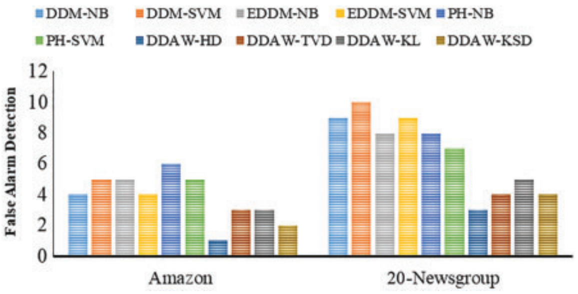

Figure 3: Comparison of drift detection methods with DDAW versions based on the false alarm on Amazon and 20-Newsgroup datasets

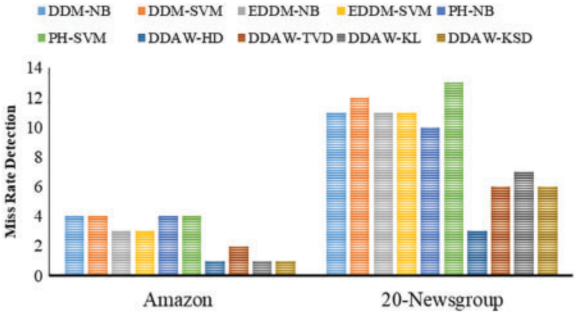

Figure 4: Comparison of drift detection methods with DDAW versions based on miss rate on Amazon and 20-Newsgroup datasets

Our strategy is more favorable for 20-Newsgroup, where concept drifts are more prevalent than for Amazon, where concept drifts are rare. On 20-Newsgroups data, the missing number in DDAW-HD is lower than in the others (including all baselines and other versions of the DDAW method), which had a missing number of 2 and a false alarm rate of 3. This is due to the dependency of baselines on classifier error rates, which results in learners missing several concept drifts since they are unable to respond to the error rate in a timely and accurate manner. Besides, a drift cannot be identified until a large enough number of misclassified instances have been accumulated. However, if the occurrences of concept drifts are high, the number of misclassified examples will reduce, hence increasing the number of missed drifts. Besides, if concept drifts are uncommon, for example in Amazon data, DDAW-HD performance would be as comparable to the baselines.

On the other hand, it should be noted that the false alarm rates of DDAW models are a little higher on the 20-Newsgroup dataset than on the Amazon dataset. In 20-Newsgroup, the false alarm rates occur because some newsgroups categories are closely related, or even overlap, such as the five computer newsgroups (comp.graphics, comp.os.ms-windows.misc, comp.sys.ibm.pc.hardware, comp.sys.mac.hardware and comp.windows). Moreover, some categories can be ambiguous and easily misclassified such as comp.sys.ibm.pc.hardware and misc.forsale. The instances of comp.sys.ibm.pc. hardware category comprises of news reports about different topics related to IBM PC hardware. While the misc.forsale category focused on news articles and advertisements concerning the sales of different items. For this dataset, it was observed that most of the instances under misc.forsale category that focused on the topics related to the personal computer (PC) sales and computer configurations were misclassified as comp.sys.ibm.pchardware. This implies that the unclear topics in the dataset may be the main reason for the high false alarm rates of the DDAW model.

4.4.2 Investigation into the Classification Results with Various Drift Detection Methods

As demonstrated in Table 2, the average error rate relates to the average of all data chunks as well as the standard deviation for those error rates. In terms of average accuracy, our DDAW-HD method surpasses previous methods on both the 20-Newsgroups and Amazon shopping datasets.

Table 2: Effect of classifiers on the drift detection approaches based on error-rate

Classifier	DDM	EDDM	Page-hinkley	DDAW-HD	DDAW-KL	DDAW-TVD	DDAW-KSD
			Amazon shopping				
NB	0.25 ± 0.25	0.28 ± 0.25	0.24 ± 0.25	0.15 ± 0.25	0.15 ± 0.25	0.15 ± 0.25	0.15 ± 0.25
SVM	0.29 ± 0.30	0.29 ± 0.30	0.30 ± 0.30	0.09 ± 0.30	0.09 ± 0.30	0.09 ± 0.30	0.09 ± 0.30
			20-Newsgroups				
NB	0.43 ± 0.21	0.44 ± 0.24	0.69 ± 0.13	0.36 ± 0.20	0.15 ± 0.25	0.15 ± 0.25	0.15 ± 0.25
SVM	0.46 ± 0.39	0.46 ± 0.39	0.75 ± 0.22	0.39 ± 0.41	0.09 ± 0.30	0.09 ± 0.30	0.09 ± 0.30

Based on Table 2, the DDAW-HD approach minimizes the error rate in 20-Newsgroups to 26.76%, which significantly outperforms the alternatives. This is due to the fact that DDAW-HD has fewer missed drifts and false alarms, resulting in a lower classification error. When DDAW-HD detects a concept drift, rather than recreating the model on the most recent chunk, it will incrementally train the classifier and make a prediction based on its incoming data, resulting in a very low classification error. On Amazon, where concept drift is uncommon, DDAW-HD performs similarly to baselines. The DDAW-HD method outperforms the competing algorithms in every scenario.

Table 2 also includes the standard deviation for error rates, and it is clear that both our models and baselines have considerable deviations. In an offline setting, the deviation is derived by combining multiple running outcomes dedicated to the same dataset. By comparing several data chunks, the deviation is computed using our data stream learning strategy. When concept drift occurs, the error rate increases, leading all algorithms, including ours and baselines, to deviate significantly. The following Fig. 5 is presented for better illustration.

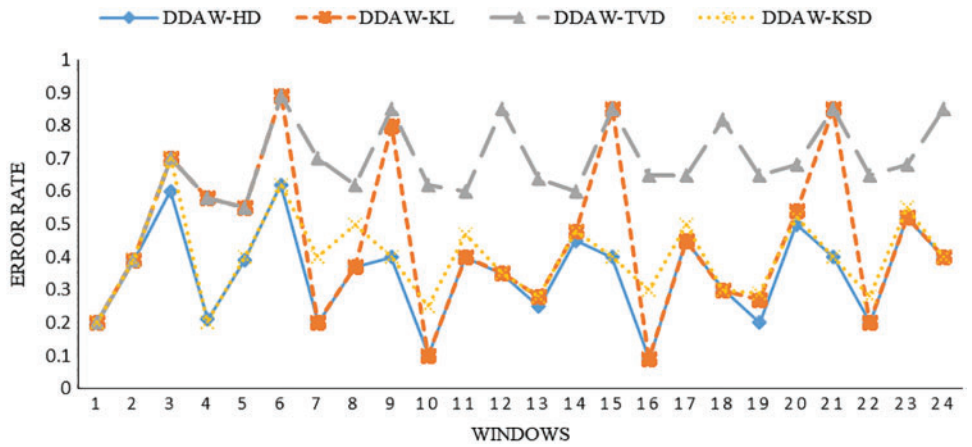

Figure 5: Concept drift detection curves on the 20-Newsgroups dataset

Fig. 5 demonstrates the effectiveness of our DDAW model using 20-Newsgroups. This figure only displays a portion of the stream data and a few examples of drift detection points. Moreover, given the experimental outcomes of the DDM, EDDM, and PageHinkley baselines are comparable, the DDAW versions in Fig. 5 are presented for better clarification.

As shown in Fig. 5, it is observed that DDAW-HD and DDAW-KSD can detect concept drift accurately at the 5th chunk, but DDAW-KL and DDAW-TVD missed the detection. Similarly, DDAW-HD and DDAW-KL can correctly detect concept drift at the 10th chunk, while DDAW-TVD and DDAW-KSD couldn’t. Furthermore, only DDAW-HD can accurately detect the concept drift at points 12th and 18th. Except for DDAW-TVD, all methods can detect the drift at point 17. Meanwhile, DDAW-HD has the lowest error rate across the board.

To examine the efficacy of our DDAW model on Amazon, which has a fewer number of data chunks (eight windows) with fewer concept drifts, a part of the data stream is presented, as well as drift detection points (5th, 10th, 12th) in Fig. 6. Similarly, it is observed that the performances of the baselines DDM, EDDM, and PageHinkley are non-significant, therefore, only the DDAW versions are shown in Fig. 5 for clarity.

As shown in Fig. 6, at point 4th chunk, DDAW-HD, DDAW-KSD, and DDAW-KL can successfully detect the concept drift, but DDAW-KSD misses it. At point 6th chunk, only DDAW-HD and DDAW-KL can successfully detect the drift correctly, but DDAW-TVD and DDAW-KSD could not detect the drift correctly. Further, at point 7th only DDAW-TVD could not detect the drift correctly.

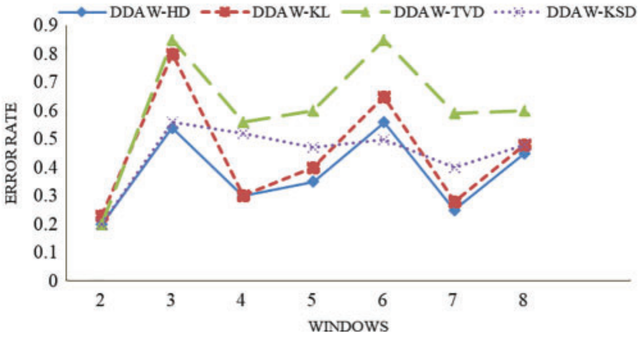

Figure 6: Concept drift detection curves on Amazon shopping dataset

The performance of the DDAW-HD method is shown better compared to the DDAW-KL, DDAW-TVD, and DDAW-KSD in terms of concept drift detection. This is presented in Figs. 5 and 6. Therefore, this proves the robustness of Hellinger distances as a concept drift measurement between distributions for univariate or multivariate data. To sum it up, the experimental findings proved that our DDAW-HD method is superior in detecting drifts sufficiently with a low missing rate, as well as adapting to new data chunks more rapidly and with a lower error rate.

4.4.3 Model Sensitivity to Ensemble Size

The ensemble size is one of the important variables that have an impact on our model. Similar to [24], the sensitivity of our proposed model towards this variable is evaluated by exploring the impacts of different sizes of ensemble k which was varied (k = 5, 7, 10, 12, 15, 18, 20) to see how it affects the performance of our model. Figs. 7 and 8 show the experimental results on all the DDAW variants, namely DDAW-HD, DDAW-KL, DDAW-TVD, and DDAW-KLS respectively while using different sizes of the ensemble on Amazon and 20-Newsgroups datasets. Each curve demonstrates the relationship between the size of the ensemble and the accuracy of classification. As can be observed from Figs. 7 and 8, the accuracy of classification increases when the ensemble has more members of classifiers. However, compared to all other variants, DDAW is not much affected by the size of the ensemble. Thus, in our experiment, since there is no strong dependency upon the size of the ensemble in terms of accuracy, k = 10 was selected as the default value for our proposed model.

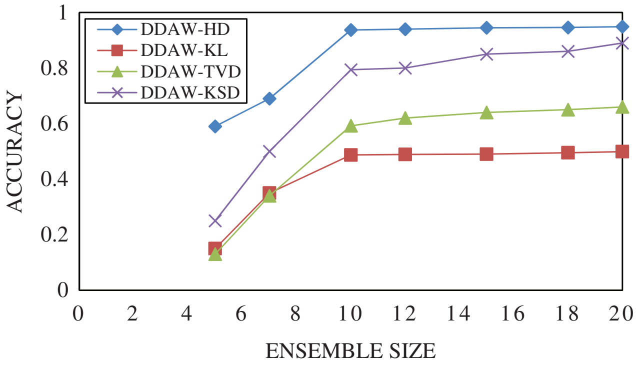

Figure 7: Results of the varying size of ensemble with DDAW variants on 20-Newsgroup dataset

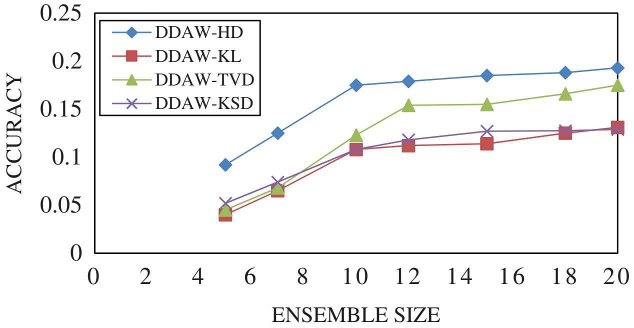

Figure 8: Results of the varying size of ensemble with DDAW variants on Amazon dataset

5 Conclusion

Concept drift detection methods provide various potentials to find possible drifts in the underlying data distribution, based on whether the change is determined through monitoring the error rates of classification models or by comparing data distributions using different similarity measures. However, most of the existing methods depend on classification errors which surfer the low-performance problems. To further improve the classification accuracy and the drift detection performance, a novel ensemble classifier is proposed which combines a number of classifiers to simultaneously discover and detect concept drifts while classifying user sentiments to improve accuracy. To achieve this, 1) a drift detection method is developed to provide timely responses towards detecting concept drift of user sentiments. In addition, a two-window technique is used to detect concept drift. 2) A number of dissimilarity measures such as Kullback–Leibler divergence, Kolmogorov–Smirnov statistic, Total variation distance, and Hellinger distance which are independent of classifiers' performances are investigated to quantify concept drift between two consecutive windows, which are known as reference window and current window. 3) a novel DDAW framework for adaptive ensemble classification is developed, considering the application of concept drift to increase the ensemble classifier's predictive performance. Through a series of experiments, it is observed that our DDAW technique that uses Hellinger distance (DDAW-HD) achieves better results compared to the other predictive models. This translates to the benefits of Hellinger distance as a distance measure to quantify the degree of changes between data distributions. This also confirms some of the previous findings suggesting the robustness of the Hellinger distance against the other distance measures and further indicates the benefits of distance measures in measuring the concept drifts compare to the error rates models. In addition, it is believed that detecting and handling concept drift in user opinions and preferences has great potential to improve sentiment classification performance. Future research will explore more of the larger-scaled datasets and other heuristic drift detection methods with other classification models to improve the performances of the concept drift detection and sentiment classification models.

Funding Statements: The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through Large Groups (Project under Grant Number (RGP.2/49/43)).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] C. C. Aggarwal, "An introduction to data streams," in *Data Streams*. Boston, MA: Springer, pp. 1–8, 2007.

[2] M. M. Gaber, A. Zaslavsky and S. Krishnaswamy, "A survey of classification methods in data streams," in *Data Streams, Advances in Database Systems*, vol. 31. Boston, MA: Springer, pp. 39–59, 2007.

[3] G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts," *Maching. Learning*, vol. 23, no. 1, pp. 69–101, 1996.

[4] I. Žliobaitė, M. Pechenizkiy and J. Gama, "An Overview of Concept Drift Applications," In: N. Japkowicz and J. Stefanowski (Eds.), *Big Data Analysis: New Algorithms for a New Society, Studies in Big Data*, vol. 16, pp. 91–114, Berlin, Germany: Springer, 2016.

[5] I. Khamassi, M. Sayed-Mouchaweh, M. Hammami and K. Ghédira, "Discussion and review on evolving data streams and concept drift adapting," *Evolving. System*, vol. 9, no. 1, pp. 1–23, 2018.

[6] J. Gama, I. Zliobaite, A. Bifet, M. Pechenizkiy and A. Bouchachia, "A survey on concept drift adaptation," *ACM Computing Surveys (CSUR)*, vol. 46, no. 4, pp. 1–37, 2014.

[7] L. Du, Q. Song, L. Zhu and X. Zhu, "A selective detector ensemble for concept drift detection," *Computer Journal*, vol. 58, no. 3, pp. 457–471, 2015.

[8] A. Bechini, A. Bondielli, P. Ducange, F. Marcelloni and A. Renda, "Addressing event-driven concept drift in Twitter stream: A stance detection application," *IEEE Access*, vol. 9, pp. 77758–77770, 2021.

[9] F. A. Pinage, E. M. dos Santos and J. M. P. da Gama, "Classification systems in dynamic environments: An overview," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 6, no. 5, pp. 156–166, 2016.

[10] R. N. Gemaque, A. F. J. Costa, R. Giusti and E. M. dos Santos, "An overview of unsupervised drift detection methods," *Wiley Interdisciplinary Review: Data Mining and Knowledge Discovery*, vol. 10, no. 6, pp. e1381, 2020.

[11] J. Zenisek, F. Holzinger and M. Affenzeller, "Machine learning based concept drift detection for predictive maintenance," *Computers & Industrial Engineering*, vol. 137, pp. 106031, 2019.

[12] D. R. de L. Cabral and R. S. M. de Barros, "Concept drift detection based on fisher's exact test," *Information Sciences (Ny).*, vol. 442–443, pp. 220–234, 2018.

[13] J. Xuan, J. Lu and G. Zhang, "Bayesian nonparametric unsupervised concept drift detection for data stream mining," *ACM Transactions on Intelligent Systems and Technology*, vol. 12, no. 1, pp. 1–22, 2021.

[14] R. de Almeida, Y. M. Goh, R. Monfared, M. T. A. Steiner and A. West, "An ensemble based on neural networks with random weights for online data stream regression," *Soft Computing*, vol. 24, no. 13, pp. 9835–9855, 2020.

[15] A. Roy, "A classification algorithm for high-dimensional data," *Procedia Computer Science*, vol. 53, pp. 345–355, 2015.

[16] R. Van Camp, "Using diversity ensembles with time limits to handle concept drift," in *Conf. Proc.-IEEE Southeastcon*, Florida, US, pp. 1–6, 2018.

[17] B. Krawczyk and A. Cano, "Adaptive ensemble active learning for drifting data stream mining," in *IJCAI Int. Joint Conf. on Artificial Intelligence*, Macao, China, pp. 2763–2771, 2019.

[18] A. Verdecia-Cabrera, I. F. Blanco and A. C. P. L. F. Carvalho, "An online adaptive classifier ensemble for mining non-stationary data streams," *Intelligent Data Analysis*, vol. 22, no. 4, pp. 787–806, 2018.

[19] M. Al-Ghossein, P. A. Murena, T. Abdessalem, A. Barré and A. Cornuéjols, "Adaptive collaborative topic modeling for online recommendation," in *RecSys 2018–12th ACM Conf. on Recommender Systems*, Vancouver, Canada, pp. 338–346, 2018.

[20] Y. Zhang, G. Chu, P. Li, X. Hu and X. Wu, "Three-layer concept drifting detection in text data streams," *Neurocomputing*, vol. 260, pp. 393–403, 2017.

[21] J. Gama, R. Sebastião and P. P. Rodrigues, "On evaluating stream learning algorithms," *Maching Learning*, vol. 90, no. 3, pp. 317–346, 2013.

[22] J. Han, W. Zuo, L. Liu, Y. Xu and T. Peng, "Building text classifiers using positive, unlabeled and 'outdated' examples," *Concurrency and Computation: Practice and Experience*, vol. 28, no. 13, pp. 3691–3706, 2016.

[23] M. Jain, G. Kaur and V. Saxena, "A K-means clustering and SVM based hybrid concept drift detection technique for network anomaly detection," *Expert Systems with Applications*, vol. 193, pp. 116510, 2022.

[24] Y. Sun, H. Shao and S. Wang, "Efficient ensemble classification for multi-label data streams with concept drift," *Information*, vol. 10, no. 5, pp. 158, 2019.

[25] J. Montiel, R. Mitchell, E. Frank, B. Pfahringer, T. Abdessalem *et al.,* "Adaptive XGBoost for evolving data streams," in *Proc. of the Int. Joint Conf. on Neural Networks*, Glasgow, UK, pp. 1–8, 2020.

[26] M. Arya and C. Choudhary, "Improving the efficiency of ensemble classifier adaptive random forest with meta level learning for real-time data streams," in *Advances in Intelligent Systems and Computing*. Singapore: Springer, pp. 11–21, 2020.

[27] S. Kumar, R. Singh, M. Z. Khan and A. Noorwali, "design of adaptive ensemble classifier for online sentiment analysis and opinion mining," *PeerJ Computer Science*, vol. 7, pp. e660, 2021.

[28] D. Margaris and C. Vassilakis, "Exploiting rating abstention intervals for addressing concept drift in social network recommender systems," *Informatics*, vol. 5, no. 2, pp. 21, 2018.

[29] L. Liu, N. Japkowicz, D. Tao and Z. Liu, "Learning with concept drift detection based on sub-concepts from k time sub windows," *Journal of Internet Technology*, vol. 21, no. 2, pp. 565–577, 2020.

[30] I. Goldenberg and G. I. Webb, "Survey of distance measures for quantifying concept drift and shift in numeric data," *Knowledge and Information Systems*, vol. 60, no. 2, pp. 591–615, 2019.

[31] T. Weerasooriya, N. Perera and S. R. Liyanage, "A method to extract essential keywords from a tweet using NLP tools," in *16th Int. Conf. on Advances in ICT for Emerging Regions, ICTer 2016-Conf. Proc.*, Negombo, Sri Lanka, pp. 29–34, 2017.

[32] F. Chu and C. Zaniolo, "Fast and light boosting for adaptive mining of data streams," in *Pacific-Asia Conf. on Knowledge Discovery and Data Mining*, Berlin, Heidelberg, Springer, pp. 282–292, 2004.

[33] F. G. D. Costa, F. S. L. G. Duarte, R. M. M. Vallim and R. F. D. Mello, "Multidimensional surrogate stability to detect data stream concept drift," *Expert Systems with Applications*, vol. 87, pp. 1339–1351, 2017.

[34] S. Misra, D. Biswas, S. K. Saha and C. Mazumdar, "Applying Fourier inspired windows for concept drift detection in data stream," in *2020 IEEE Calcutta Conf., CALCON 2020-Proc.*, Kolkata, India, pp. 152–156, 2020.

[35] Y. Sun, Z. Wang, Y. Bai, H. Dai and S. Nahavandi, "A classifier graph based recurring concept detection and prediction approach," *Compututational Intelligence and Neurosciences*, vol. 2018, pp. 13, 2018.

[36] J. E. Contreras-Reyes and R. B. Arellano-Valle, "Kullback-leibler divergence measure for multivariate skew-normal distributions," *Entropy*, vol. 14, no. 9, pp. 1606–1626, 2012.

[37] Z. Wang and W. Wang, "Concept drift detection based on Kolmogorov–Smirnov test," in *Artificial Intelligence in China*. Singapore: Springer, pp. 273–280, 2020.

[38] J. McAuley and J. Leskovec, "Hidden factors and hidden topics: Understanding rating dimensions with review text," in *RecSys 2013-Proc. of the 7th ACM Conf. on Recommender Systems*, Hong Kong, pp. 165–172, 2013.

[39] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard *et al.*, "The Stanford CoreNLP natural language processing toolkit," in *Proc. of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, Baltimore, Maryland, pp. 55–60, 2014.

[40] M. Baena-García, J. del Campo-Ávila, R. Fidalgo, A. Bifet, R. Gavaldà *et al.*, "Early drift detection method," in *Fourth Int. Workshop on Knowledge Discovery from Data Streams*, New York, vol. 6, pp. 77–86, 2006.

[41] Y. Sakamoto, K. Fukui and D. Nicklas, "Concept drift detection with clustering via statistical change detection methods," in *Seventh Int. Conf. on Knowledge and Systems Engineering*, IEEE, New York, NY, USA, pp. 37–42, 2015.