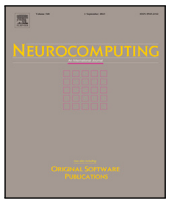
Neurocomputing 617 (2025) 128933

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Novelty-aware concept drift detection for neural networks

Dan Shang, Guangquan Zhang, Jie Lu *

Australian Artificial Intelligence Institute, Faculty of Engineering and Information Technology, University of Technology Sydney, Australia

ARTICLE INFO

Communicated by N. Zeng

Keywords:
Concept drift detection
Stream data mining
Neural network

ABSTRACT

Neural network models are widely adopted in real-world applications for processing streaming data. However, these applications often face challenges in terms of accuracy degradation, caused by changes in the data distribution of the stream data compared to the training data. Two underlying reasons contribute to these changes. The first, known as the concept drift problem, occurs when there is a change in the correlation between the input data and the prediction output, making the models trained on the training data no longer suitable for the new data. The second reason, known as the novelty problem, arises when real-world data contains unexpected data categories that were not present in the training data, resulting in incorrect predictions. The research community has divided into different groups and each developed various methods to detect either concept drift or novelty distribution changes. However, these methods only address one aspect of the problem and are unable to distinguish between them. This leads to an inappropriate allocation of model maintenance resources, including the high cost of model retraining and the acquisition of true label data. In this study, we aim to address this gap by proposing a novel concept drift detection method that is capable of distinguishing between known labeled concept drift and novelty. Our method is also more efficient than existing drift detection methods, making it suitable for applications on neural networks.

1. Introduction

Stream data is being generated from multiple applications in many different scenarios, for example, stock market quotes, outputs from various equipment sensors, social media feeds. Thus, stream data mining attracts lots of attentions of researchers. Unlike traditional machine learning problems, stream data present particular challenges to machine learning models. Particularly, when applying a machine learning model to real-world data, one of the challenges is the data distribution of the stream data often changes compared to the training data. Such unexpected distribution discrepancy of stream data hampers prediction accuracy. Different groups in research community have attempted to address this issue in different directions.

One group noticed that the distribution changes occur in the correlation between the input data and the prediction output [1]. The models trained on the training data no longer suitable for the new data generated in the real-world applications. This is known as the concept drift problem and is well studied in research community [2]. The widely adopted solution is to detect the concept drift and then retrain the models with latest data to fit the distribution change. The core algorithm is the concept drift detection method, which can be implemented in different strategies. Error rate-based detection methods [3,4] monitor the models output and check the prediction values against true label data. The detection algorithm raises concept drift alarms when the error rate exceeds some predefined threshold. Such methods apply univariate statistical tests on the error rate, thus are very efficient and have low resource footprint. Also, they only reports alarms on actual accuracy drops, and ignore irrelevant distribution changes, thus avoid unnecessary model maintenance. However, the major limitation of such methods is that they are based on true label data, which can be unavailable in real-world stream data applications. Another strategy is to apply two-sample testing on the input data to detect distribution changes. These methods [5,6] need to use multivariate two-sample tests since the input data often has high dimension. The most noticeable advantage of these methods is that they only consume input data, and do not require true label data. However, they often suffer from poor performance due to their calculation on the distances between two high dimensional data samples. They only alert concept drifts according to the distribution tests. They cannot distinguish distribution changes caused by new emerging classes from distribution changes caused by the correlation change. Such distinguishing is valuable knowledge for further concept drift understanding or model adaptation.

Researchers in another group approach the data distribution change problem by focusing on the unknown concepts of the data, contrary to the known concepts in the concept drift problem [7]. They emphasize the scenario where the incorrect predictions are not cause by

* Corresponding author.
E-mail addresses: dan.shang@student.uts.edu.au (D. Shang), guangquan.zhang@uts.edu.au (G. Zhang), jie.lu@uts.edu.au (J. Lu).

https://doi.org/10.1016/j.neucom.2024.128933
Received 5 August 2023; Received in revised form 21 October 2024; Accepted 13 November 2024
Available online 22 November 2024

the correlation change of the input features and predictions, but the existence of unseen prediction categories that are not covered by the training data. Such unknown distribution is named as "novelty". Novel problem is a major challenge in neural network applications, such as video surveillance, object recognition and planetary exploration, where unexpected objects or subjects appear in footage or image. In literature, the goal of novelty detection usually means detecting the out-of-distribution samples one by one. It is also referred as to open set recognition, outlier detection or anomaly detection. They mainly target on testing samples as erroneous, fraudulent, or malicious. Novelty detection methods cannot rely on the error rates, since the predictions made by the models are no longer trustworthy. Coincidentally when comparing to concept drift detection methods, novelty detection methods also resolve to high dimensional two-sample tests to estimate the density difference between training data distribution and test data distribution. Some methods use parametric tests, for example, a multivariate Gaussian distribution [8,9] or more complex tests, such as mixed Gaussian distribution [10,11], and Poisson distribution [12], to measure the distance between the training and the test data. Other methods, when it is difficult to anticipate the real-world data distribution, use non-parametric tests, such as high dimensional kernel density estimation [13], to measure density difference of the training data and test data. Similar to concept drift detection methods that use high dimensional two-sample tests, novelty detection methods also suffers from performance issues, especially for computer vision tasks based on neural networks. In addition to recognizing novelty or new concepts, an effective online learning strategy should also be capable of detecting changes that occur within known or normal concepts. It is crucial for the model maintainer to determine whether the decrease in model accuracy is a result of concept drift or novel data. This information is essential for determining the most effective maintenance strategy and for avoiding the high cost of acquiring ineffectual true label data that is unrelated to the distribution change. The current state of the art methods in this area has limitations that prevent a full understanding of the underlying causes of data distribution changes.

Although both concept drift problem and novel problem are well addressed in the literature and plethora methods have been developed, there remains a considerable gap in the solutions for these two problems. One of the key challenges in this field is the inability of existing methods to distinguish between concept drift and novelty. This results in a situation where concept drift detection methods may raise an alarm in the presence of novel data, while novelty detection methods may report a novelty alarm in the presence of concept drift. This lack of differentiation can lead to incorrect maintenance actions and an inappropriate allocation of resources. Moreover, the simultaneous application of both concept drift detection and novelty detection methods is not an effective solution, as they both raise alarms in response to any change in the data distribution. This can result in an inefficient use of resources, such as unnecessary model retraining and data acquisition.

In this study, we aim to address this gap by proposing a novel concept drift detection method that is capable of detecting both concept drift and novelty, and it also distinguishes between the two types of distribution changes by producing the type of change. This information is crucial for model maintainers, who can then adopt the most appropriate actions, such as requesting updated labeled data or investigating new data categories, to keep the model updated. Instead of defining novelty as examples that do not fit a certain data distribution, we view novelty as a concept represented by a cluster of new class examples. In our paper, novelty is not merely an outlier, but rather a distinct concept within the data. Novelty samples reflected in radial base distances present higher values. We leverage this property and removing a percentile value such that higher values of radial distances, which correspond to the novelty samples, can be excluded. After removing the predefined percentile of samples with higher radial distances, we compare distributions of the remaining samples. If the distribution change are caused by novelty, the remaining samples should have the same distribution. Otherwise, we consider that the distribution change are caused by concept drift. The main contributions are:

- Propose a method that can detect and distinguish stream data distribution change caused by concept drift and novelty.
- The method is designed to have low memory and computation footprint with high dimensional input, which make suitable for neural network models.
- Introduce a theoretical foundation that for the first time make it possible to analyze concept drift problem and novelty problem in one framework.

The remaining parts are organized as follows. Section 2 reviews the established literature on concept drift detection and novelty detection. In Section 3, we first present a formal definition of the distribution change phenomenon in stream data mining, and its two possible root causes, namely concept drift and novelty. After that, we introduce the distance measurement which is the foundation of our two sample test statistic. Finally, we present the novelty-aware concept drift method and its supporting theories. Section 4 illustrates effectiveness of our proposed method and the computational performance on high dimensional data in both concept drift and novelty scenarios. Section 5 summarizes our work and directions for future study.

2. Related work

This literature review covers the research problem and existing solutions relevant to our aim of developing a novelty-aware concept drift detection method for neural network classification tasks. A rich body of independent literature already exists for concept drift detection and novelty detection. Hence, in this section, we review the representative works from these two distinguished research areas and analyze their relationships to our method.

2.1. Concept drift detection

Concept drift in machine learning refers to the shift in the distribution of data that results in decreased model performance [2]. Concept drift is a well-known issue in the field of machine learning research that can have a significant impact on the performance of models. The problem arises when the underlying distribution of the data changes over time, making it difficult for the model to generalize to new data [14]. This can result in a decrease in accuracy and an increase in the number of errors. In online classification tasks, the concept drift problem arises when prediction performance deteriorates over time due to changes in the distribution of the data. These changes can be due to either a shift in the features or the target concept, causing either virtual or real drift [15]. The latter, also known as concept drift, is a major challenge in online machine learning applications.

Formally, given a classification task with a feature vector $X \in \mathbb{R}^d$ and class labels $y \in \{1 \dots c\}$, an initial set of samples with known labels is used to train the classifier model $P(y|X)$. Concept drift is defined as that there exists a time t_0, $P_{t<t_0}(X, y) \neq P_{t\geq t_0}(X, y)$. Since $P(X, y) = P(X) \cdot P(y|X)$, two types of distribution changes can be observed: (1) virtual drift, where $P_{t<t_0}(X) \neq P_{t>t_0}(X)$, while $P_{t<t_0}(y|X) = P_{t>t_0}(y|X)$; and (2) real drift or concept drift, where $P_{t<t_0}(X) = P_{t>t_0}(X)$, while $P_{t<t_0}(y|X) \neq P_{t>t_0}(y|X)$. Concept drift is one of the root causes of performance degradation in online machine learning applications.

To address this problem, researchers have developed two basic strategies. The first strategy involves using incremental classifiers. These models are trained using the latest labeled data, allowing them to continuously adapt to changes in the underlying distribution of the data. This approach is effective in dealing with concept drift as it allows the model to constantly update itself and maintain its accuracy. However, this approach requires a continuous flow of labeled data and is dependent on the underlying classifier used. The second strategy is to use a standalone concept drift detection algorithm. This approach is classifier-independent and has a wider application scope. The idea behind this approach is to monitor the distribution of the data stream and trigger model maintenance when a concept drift occurs. This can

be done by comparing the distribution of the new data to that of the historical data and detecting any changes in the distribution. This information can then be used to retrain the model and ensure that it continues to perform well even when the distribution of the data changes. The main focus of this article is on drift detection, which can be used as an external tool to supervise the data stream and help maintain the performance of the model.

In the early days of machine learning research, error rate-based drift detection algorithms were widely used, due to their simplicity and efficiency. For example, in online classification applications, an increase in prediction error rate is considered as an indicator of concept drift, and this triggers a classifier update. The drift detection method (DDM), first introduced in [16], is considered a seminal work in this area and lays the foundation for this approach. DDM operates by monitoring the prediction error rate of a classifier based on instances in the most recent time window. If the error rate reaches a pre-defined warning level, a new classifier is trained, and if the error rate exceeds the drift level threshold, it replaces the existing classifier. Several extensions to DDM have been developed to better address specific types of drift, such as the early drift detection method (EDDM) [17], dynamic extreme learning machine (DELM) [18], and Hoeffding's inequality-based drift detection method (HDDM) [19]. A more recent work, EWMA [20], has employed an exponential weighted moving average (EWMA) chart to test the significance of changes in error rates, so as to reduce false alarms. The EWMA chart is a widely used quality control statistic. Despite the many advances in error rate-based algorithms, they still have a major limitation — they cannot describe the concept drift. As a result, they only provide drift alarms, but not any additional information for classifier maintenance, beyond simply detecting the drift.

In order to detect distribution changes, multivariate two-sample tests can be applied. These tests compare the distribution of instances in the training data with the most recent data. In order to perform these tests efficiently, researchers have proposed different techniques. Reis et al. proposed a fast drift detection method by extending the univariate Kolmogorov–Smirnov (KS) test to higher dimensions [21]. In their method, data instances are organized using high-dimensional random trees. This allows the tree structure to be updated in $O(\log N)$ time when a new instance arrives. After updating the tree, the KS statistic can be computed in $O(1)$ time, resulting in a sublinear performance suitable for online data stream applications. On the other hand, Rosenbaum et al. proposed a new statistic for multivariate two-sample tests [22]. This method counts the number of different types of pairs formed by optimal non-bipartite matching between instances from the two samples. Unlike other methods, this approach operates independently of the distribution and the exact distribution of the statistic is provided. However, computing the optimal matching has a complexity of $O(N^3)$, making it unfeasible for real-time applications.

Finally, the third category of concept drift detection algorithms is instance-based distribution tests. These algorithms compare the distributions of instances, rather than features or error rates, to detect drift. The key idea behind this category is that instances from different time windows are representative of different concepts, and a shift in the distribution of instances reflects a concept drift. The theoretical foundation for this approach was first proposed by Kifer et al. [23]. They provided statistical guarantees for the test and introduced a resampling procedure to estimate the significance threshold of the test statistic. This approach is considered distribution-free as it only requires that the instances in the stream data are generated independently. However, the exact distribution of the statistic can often be difficult to derive, and as a result, several different methods have been proposed that follow this framework. These methods differ mainly in their partitioning strategies and the statistics used to count regional differences. For example, the k-dimensional quad-tree (KdqTree) algorithm [24] uses a tree structure to partition a high-dimensional sample space into hypercubes of similar size. The Kullback–Leibler divergence is then used to sum the counting differences for each cell. To address performance issues, bootstrapping is used to estimate the significance threshold, and as a result, updating the tree has a time complexity of $O(\log N)$ when new data arrives. One popular instance-based drift detection method is the Page–Hinkley test [25]. The Page–Hinkley test is a statistical process control method and is widely used in quality control applications. The method monitors the cumulative sum of the deviations between the actual and expected mean of instances and triggers a drift alarm when the cumulative sum exceeds a predefined threshold. Another instance-based method is the two-phase change detection method [26]. The two-phase method starts with a rough estimation of the mean and covariance of the data stream, followed by an online adjustment when a drift is detected. When the mean and covariance change significantly, the method switches to the second phase, which starts with a full recalculation of the mean and covariance. The method is effective and efficient, but it relies on the Gaussian assumption of the data distribution. Another approach was proposed by Lu et al. [27] who introduced a drift detection method based on a competence model and case-based reasoning theories. The competence of a set of labeled instances or the case base describes the model's ability (i.e., certainty) to predict an unknown instance using k-nearest-neighbor (kNN) rules. The algorithm measures the competence difference to compare the distributions of the two samples and computes the competence for a given case base by partitioning the sample space into overlapping hyperspheres. The significance threshold is determined by permuting the sample data multiple times to estimate the variance of the competence. Although this method is accurate, it can be computationally expensive, as computing the competence alone has quadratic complexity, which is then multiplied by the permutation process. To address this drawback, Lu et al. [28] presented a method to mitigate this drawback by introducing an instance reduction procedure based on the case base. This method helps to reduce the complexity of the algorithm, making it more suitable for stream data processing.

2.2. Novelty detection

Novelty detection in machine learning and neural network research has been a topic of interest in recent years due to the increasing need for robust and accurate methods [29] to identify unusual or unseen patterns in data. The focus on this area of research stems from its wide range of applications [30] in various domains such as computer vision, speech recognition, cybersecurity, video surveillance, object recognition, and planetary exploration.

One of the popular approaches for novelty detection is to model the distribution of the normal data [10]. They assume that the novelty data, deviating from the expected distribution, has lower likelihood under the modeled distribution than the normal data. Different distribution modeling methods were used in literature, including parametric [8] and non-parametric tests [31]. Parametric tests choose as a priori some predefined distribution, such as multivariate Gaussian distribution, mixed Gaussian distribution, or Poisson distribution. The normal data are considered to follows a specific distribution. These methods flag instances that are far away from the training data as novelties [8]. However, when the real-world data distribution is unknown or difficult to anticipate, non-parametric tests, such as high dimensional kernel density estimation [32], can be used to measure the density difference between the training and test data distributions. These methods do not make any assumptions about the data distribution and can be more robust in detecting novelties in complex and diverse data sets. Despite the advantages of the two-sample tests, the performance of novelty detection methods based on these tests can be challenging in high-dimensional data [33], such as computer vision tasks based on neural networks. The high dimensionality of the data can make it difficult to accurately estimate the density difference, leading to a higher rate of false positive or false negative detection. To address this issue, researchers have proposed various techniques, such as dimensionality reduction, feature selection, and ensembles of novelty detection models, to improve the performance of novelty detection methods [34].

Neural network are often used to enhance the performance of distribution based novelty detection in high dimensional space [35]. The high representation quality of deep neural networks has significantly improved the performance of classic density estimation methods. Autoencoder (AE) and Variational Autoencoder (VAE) based models are two of the most widely used deep learning techniques for novelty detection [36,37]. Autoencoders were first introduced in the field of unsupervised learning [38], and they learn efficient representations of unlabeled data by reconstructing the input from the latent embedding. On the other hand, Variational Autoencoders (VAEs) [39] are a generative model that encodes input images into latent vectors under the Gaussian distribution. The advantage of VAEs over autoencoders is that they not only reconstruct the input data but also generate new samples by sampling from the learned Gaussian distribution. The learned deep representation from encoding can be used an representation of the higher dimensional the input. Methods based on them can perform unsupervised anomaly detection [36]. Several studies have demonstrated the effectiveness of AE/VAE-based models for novelty detection in various domains, such as computer vision and speech recognition [40]. For instance, in computer vision, AE/VAE-based models have been used for anomaly detection in images and video sequences [41]. These models have also been applied to speech recognition tasks, where they have been used for detecting unusual or unexpected speech patterns [37].

GANs [42] are a type of deep generative model that consist of two neural networks: a generator network and a discriminator network. Unlike Autoencoder, while GANs have been effective in generating new samples. they do not have the encoder to produce corresponding embedding for a given sample. To overcome this challenge, the ADGAN [43] (Anomaly Detection GAN) method has been proposed. This method searches for a good representation in the latent space for a given sample and if such a representation is not found, the sample is deemed novelty. However, this method may be with high computation cost, especially when dealing with large data sets.

Previous works have shown that human perception of images is largely based on low-frequency components, while deep neural networks such as convolutional neural networks (CNNs) can heavily rely on high-frequency components for decision-making. To address this issue, researchers have proposed methods to suppress the influence of high-frequency components, such as CNN kernel smoothing and spectrum-oriented data augmentation. These methods aim to improve the performance of CNNs in anomaly detection by reducing their reliance on high-frequency components. In addition to these methods, recent works have also found that adversarial attacks on low-frequency components are difficult to detect, leading to the proposal of methods that target the phase spectrum. These frequency-based methods focus mainly on sensory anomaly detection, especially on detecting adversarial examples.

3. Methodology

The problem we aim to solve is to differentiate concept drift and novelty further when distribution change has been detected. In this section, we first present a formal definition of the distribution change phenomenon in stream data mining, and its two possible root causes, namely concept drift and novelty. After that, we introduce the distance measurement which is the foundation of our two sample test statistic. Finally, we present the novelty-aware concept drift method and its supporting theories.

3.1. Problem definition

Data stream mining and statistics research are focused on analyzing the continuous flow of instances generated in real-time. The sequence of data generated in a stream is denoted as $\{s_1, s_2, \dots, s_i, \dots\}$, $i \in \mathbb{N}^+$. The size of the sequence may be unbounded and infinite, making it a challenging task to analyze.

In unsupervised data stream mining, each instance is represented as a d-dimensional vector $s_i = [x_i^1, x_i^2, \dots, x_i^d]$, where $x_i \in \mathbb{R}^d$ represents the jth feature of the ith sample s_i. On the other hand, in supervised classification tasks, each instance $s_i = [x_i^1, x_i^2, \dots, x_i^d, y_i]$ is represented as a pair (X_i, y_i) of input X and classification output Y, with the last component y_i representing the assigned label and taking values from a finite set of classes with a finite number of values $Y = \{c_1, c_2, \dots, c_m\}$. $m \in \mathbb{N}^+$, where $m \in \mathbb{N}^+$ is a finite number. In a probabilistic setting, each observation in the data stream s_i can be considered as a pair of random variables (X_i, y_i) drawn from the sample space $S \subset \mathbb{R}^d \times \mathbb{R}^1$. $X_i = [x_i^1, x_i^2, \dots, x_i^d]$ takes values in $X \subset \mathbb{R}^d$, and $y_i \in Y \subset \mathbb{R}^1$. The underlying process $P(X, y)$ that generates (X_i, y_i) can be determined by the probability of observing $P(X)$, and the posterior probability $P(y \mid X)$. The classifier, $f : X \to Y$, trained on the data sets $\{(X_i, y_i)\}_{i=1}^n$ reflects the generating process $P(X, y)$, aiming to make the probability $P(f(X_j; X_1, y_1, X_2, y_2, \dots, X_n, y_n) \neq y_j)$ as small as possible. This is based on the assumption that the data sets in the stream are independent identically distributed (i.i.d.) according to the distribution $P(X, y)$. In other words, the classifier is trained to generalize the underlying generating process and minimize the probability of observing $f(X_j; X_1, y_1, X_2, y_2, \dots, X_n, y_n) \neq y_j$.

In practical applications, the probability distribution of the underlying data generating process $P(X, y)$ is often not consistent over time. The joint probability distribution $P(X, y)$ can change and cause a decrease in the performance of previously trained classifiers $f : X \to Y$. This change can occur in two ways. Firstly, it can occur in the conditional probability of observing y given X, denoted as $P(y \mid X)$, leading to real drift, which results in an increase in the error rate due to a shift in the posterior distribution of class membership. Secondly, change can occur in the probability of observing X, denoted as $P(X)$, leading to virtual drift. In this case, only the feature space $\{X\}$ changes and does not affect the posterior distribution of class membership $P(y \mid X)$. Real drift can be caused by correlation change between input X and output y, known as concept drift, or caused by the emergence of unknown concepts, that is new values of y that do not exist in the label space of training data, known as novelty. Real drift, in particular, can cause the models to become outdated and produce incorrect results, as the posterior distribution of class membership has shifted. This highlights the importance of continuously monitoring and updating models to adapt to changes in the data distribution. On the other hand, virtual drift can be easier to handle as it only affects the feature space and not the posterior distribution. However, it is still important to detect and adapt to these changes to maintain the performance of the models. Due to the fact that in real-world stream data mining applications, true label data is often unavailable, our method is designed to works under unsupervised conditions. In classification tasks, the new incoming data sets will be clustered into several subsets according to their predicted labels y'_j. In this paper, we monitor concepts drifts by supervising the distribution change of each cluster $\{X_j : f(X_j; X_1, y_1, X_2, y_2, \dots, X_n, y_n) = y'_j\}$.

In unsupervised situations, one commonly employed approach to detect distribution changes is through the use of two sample testing methods. These methods aim to determine if two batches of data sets are from the same underlying distribution, represented by the probability density function. If the results of the two sample test are statistically significant, it is considered that a concept drift has occurred. However, it is important to note that this approach can lead to a higher rate of false alarms triggered solely by changes in the distribution between classes. This is because the change in $P(x)$ does not necessarily translate to a change in the underlying relationship between the features and the class labels. This relationship can be better understood through the Law of Total Probability $P(X) = \sum_{c_i \in Y} P(y = c_i) * P(X \mid y = c_i)$. From this equation, we can see a change in the ratio between classes can also result in a change in $P(x)$, but this does not necessarily mean that the classification accuracy is impacted. In such cases, it is important to consider the underlying changes in prior probabilities $P(y = c_i)$ and the

corresponding changes in the conditional probabilities $P(X \mid y = c_i)$. However, it is also important to note that the classification boundaries may remain unchanged even in the presence of such changes. To address this challenge, stream data mining researchers often focus their efforts on detecting changes in the distribution of $p(X \mid y = c_i)$ rather than the entire input data set. The assumption is that if the distribution of a particular class changes, it is possible that the classifier boundary will change as well. This requires the monitoring and analysis process is typically performed on a per-class basis. This approach is particularly useful in dealing with class imbalanced classification problems, where the majority class may overwhelm the minority class. By monitoring and analyzing the distributions of each class separately, the impact of the majority class on the minority class can be minimized. In practice, we typically test for changes in $P(x \mid y' = c_i)$ using statistical methods. These predicted labels are usually available without incurring additional costs, making the process of detecting changes in distribution more efficient and cost-effective.

3.2. Radial base distance

For high-dimensional data sets, reducing the dimensionality of the data sets can provide several key benefits. The first benefit is related to memory and computation resource efficiency. By reducing the dimensionality of the data sets, it is possible to store only a portion of the feature set, rather than the entire set. This reduction in the amount of data stored can help to conserve valuable memory resources and reduce the computational demands of the analysis process. The second benefit of reducing the dimensionality of high-dimensional data sets is related to model accuracy. Removing redundant features can help to improve the accuracy of models by reducing the potential for over-fitting. Over-fitting occurs when a model becomes too closely tailored to the training data, leading to poor generalization performance on unseen data. By reducing the number of features, it is possible to avoid over-fitting and to improve the accuracy of the models used in stream data mining and machine learning.

In this regard, our approach involves extracting a single feature for use as the input for detection. One popular dimension reduction method that can be used in this context is principal component analysis (PCA). PCA assumes that variance corresponds to the information content of the data set, and the method is widely used for reducing the dimensionality of high-dimensional data sets. In our approach, we choose a distance metric that has enough discrimination power to accurately represent the original high-dimensional points in memory. In contrast to other distance based methods that might choose the centroid of the data set, we instead choose a fixed point, denoted as $B = (b_1, b_2, \ldots, b_n)$. We name it as radial base, which can be defined as follows:

Definition 3.1 (*Radial Base*)**.** The radial base is a fixed point we choose, which locates in the subspace of the entire feature space. It has the following property: distance from it is more sensitive to change in the directions of the subspace.

For any n dimensional sample $X = (x_1, x_2, \ldots, x_n)$, the Euclidean distance from radial base B is named as radial distance, denoted as $d(X)$: $d(X) = \|X - B\| = \sqrt{\sum_{i=1}^{n}(x_i - b_i)^2}$

From the above equation, if we want to enlarge the sensitivity of $d(X)$ with respect to x_i the derivative along x_i, denoted as $\frac{\partial d(X)}{\partial (x_i)}$, should be made larger. $\frac{\partial d(X)}{\partial (x_i)} = \frac{x_i - b_i}{\sqrt{\sum_{i=1}^{n}(x_i - b_i)^2}}$

From the above equation, we can refer that if the $x_i - b_i$ is larger relative to the other features $x_j - b_j$, the derivative $\frac{\partial d(X)}{\partial (x_i)}$ will be larger. That means the radial distance feature d will be more sensitive to the change of x_i. The radial base should be relatively close to the data set, but the values of the features that are of primary interest in the detection process are relatively far from the mean. Algorithm 1 presents the radial base initialization process. The radial distance is the feature we perform drift detection on. This approach allows us to more effectively capture the important information contained in the data set, while reducing the memory requirements and computational demands of the analysis process.

To ensure that the radial distance feature is more sensitive to changes in the features related to a particular class, a radial distance base is chosen separately for each class. In neural networks, each class has its own feature subspace more related with it. The choice of a separate radial distance base for each class ensures that the radial distance feature is more sensitive to changes happen in that subspace. By carefully choosing the location of the radial distance base, we can control the sensitivity of the subspace and better understand the changes occurring in the data set over time. This ability to control the sensitivity for changes in the subspace provides a flexible and efficient tool for detecting changes and making informed decisions about how to adapt our models to these changes. Radial distance also has a property that will help to differentiate concept drift from novelty. For each class samples, if they have part of new class samples in it, the radial distance feature of new class samples should be larger than the existing class samples. If we remove this larger percentile radial distances, the remaining part should have the same distribution with the training data set radial distances. But if the distribution change are concept drifts, the remaining part still have different distribution with the training data set radial distances. By utilizing this property of radial distances, we can further differentiate concept drift from the novelty.

Fig. 1 illustrates an exemplary concept drift data sets, reflected on probability density function (PDF) and cumulative distribution function of radial base distances. Sub-figure (a) illustrates two-dimensional samples from training data set and test data set in different colors, generated by multivariate Gaussian distribution with different means to simulate the concept drift. The radial base distances of the samples are computed against the radial base on the dimension where the distribution change occurs, displayed as dashed line. Sub-figure (b) demonstrates the PDF variation of the radial base distances of the two sample sets. Sub-figure (c) demonstrates the CDF variation of the radial base distances of the two sample sets.

Comparatively, Fig. 2 illustrates an exemplary novelty data sets, reflected on probability density function (PDF) and cumulative distribution function of radial base distances. Sub-figure (a) illustrates two-dimensional samples from training data set and test data set in different colors, generated by multivariate Gaussian distribution. Noticeably, the distribution of the majority of the test samples (marked in orange crosses) does not change comparing to training samples. However, some test samples (marked in orange triangles) are generated by multivariate Gaussian distribution with different means and variances to simulate the novelty. The radial base distances of the samples are computed against the radial base on the dimension where the distribution change occurs, displayed as dashed line. Sub-figure (b) demonstrates the PDF variation of the radial base distances of the two sample sets. Sub-figure (c) demonstrates the CDF variation of the radial base distances of the two sample sets.

3.3. Novelty-aware concept drift detection

In our proposed method, we store the data points in an efficient manner. Specifically, we utilize a set of radial distances to represent the data points, rather than the original data points themselves. This is because distances are simply real numbers with one dimension, allowing for a significant reduction in memory usage compared to other representations. To adjust the discrimination power of the radial distances for different subspaces of the data sets, we introduce the concept of a radial base, which is a flexible and adjustable parameter. Given a specific class of data set $\{s_i^{c_k}\}$ with class label c_k, the radial distances between $s_i^{c_k}$ and the radial base B^{c_k} are computed, denoted as $dist_i^s = \left\| s_i^{c_k} - B^{c_k} \right\|$. These radial distances serve as a new representation of the data points, generating a new data set $\{dist_i^{c_k}\}$ from the original

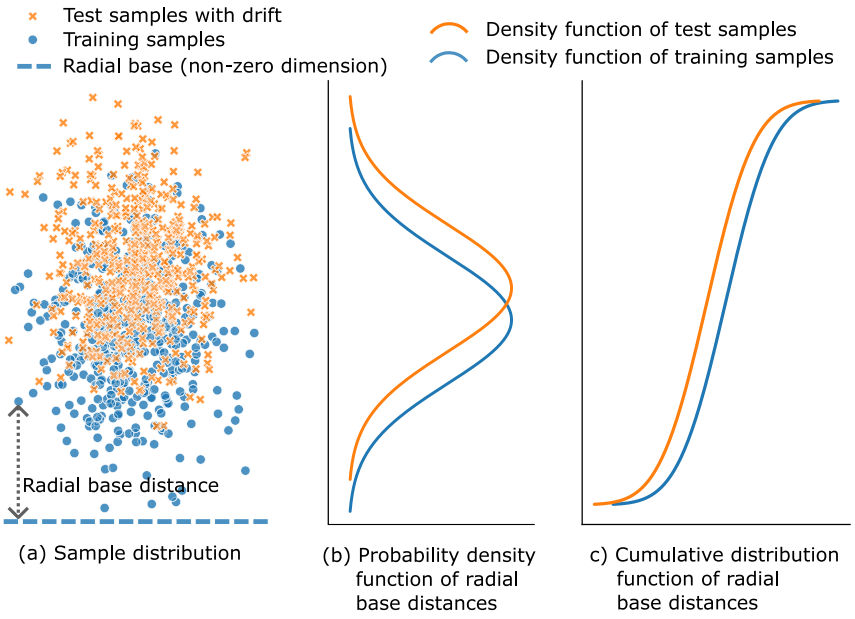


Fig. 1. Concept drift reflected on probability density function (PDF) and cumulative distribution function of radial base distances. Sub-figure (a) illustrates two-dimensional samples from training data set and test data set in different colors, generated by multivariate Gaussian distribution with different means to simulate the concept drift. The radial base distances of the samples are computed against the radial base on the dimension where the distribution change occurs, displayed as dashed line. Sub-figure (b) demonstrates the PDF variation of the radial base distances of the two sample sets. Sub-figure (c) demonstrates the CDF variation of the radial base distances of the two sample sets.

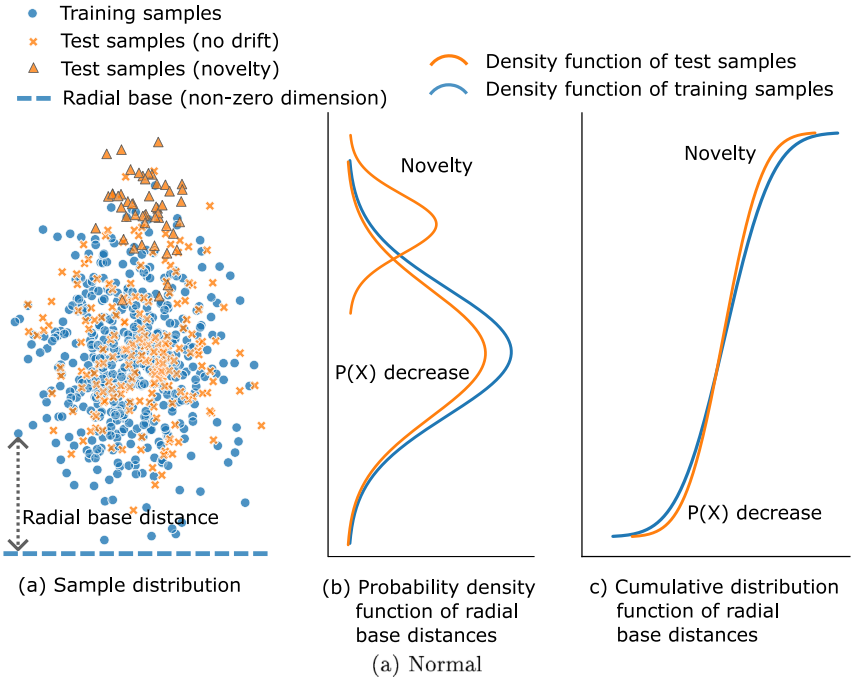


Fig. 2. Novelty reflected on probability density function (PDF) and cumulative distribution function of radial base distances. Sub-figure (a) illustrates two-dimensional samples from training data set and test data set in different colors, generated by multivariate Gaussian distribution. Noticeably, the distribution of the majority of the test samples (marked in orange crosses) does not change comparing to training samples. However, some test samples (marked in orange triangles) are generated by multivariate Gaussian distribution with different means and variances to simulate the novelty. The radial base distances of the samples are computed against the radial base on the dimension where the distribution change occurs, displayed as dashed line. Sub-figure (b) demonstrates the PDF variation of the radial base distances of the two sample sets. Sub-figure (c) demonstrates the CDF variation of the radial base distances of the two sample sets.

Algorithm 1: Radial Base Distances

input : X, training data set of a specific class;
Z^L, activation of layer L in neural network model;
γ, threshold of layer activation values, by default 0.5;
λ, integral multiplier of $\overline{Z_j^L(x)}$, by default 2.
output: radial distance base.
referential radial distances.

forall *input x in X* **do**
  get layer activations $Z^L(x)$ for input x;
  foreach *neuron j in layer L* **do**
    find $\overline{Z_j^L(x)}$ as the mean activation value of neuron j given all x_{train};
initialize vector B as radial distance base
    B=λ*(neuron j in L: $\overline{Z_j^L(x)}$);
find threshold activation value t of all values in B according to threshold γ;
foreach *value b in B* **do**
  if $b < t$ **then**
    set $b := \overline{Z_j^L(x)}$;
initialize $R :=$ empty list;
forall *input x in X* **do**
  get activation $Z^L(x)$ for input x;
  compute $Dist(Z^L(x), B)$, the distance from activation vector to base vector;
  append $Dist(Z^L(x), B)$ to R;
return *B as the radial distance base; R as the referential radial distances.*

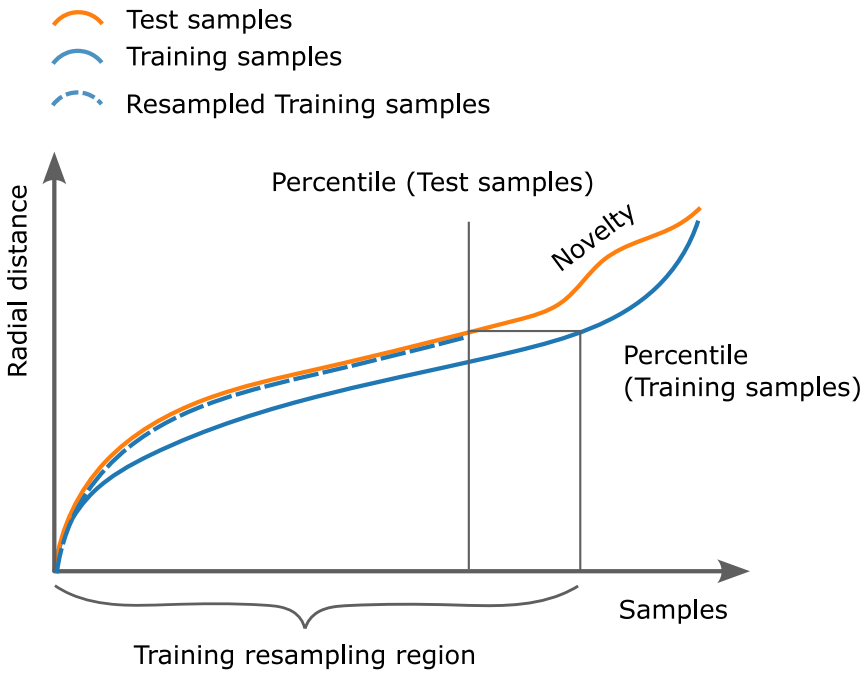
data set { $s_i^{c_k}$ }. The process is repeated for subsequent batches of data, yielding a sequence of corresponding distance data sets that represent the evolution of the original data set over time. To detect potential drifts in the data distribution, we must choose methods that are suitable for one-dimensional settings. One such method is the Kolmogorov–Smirnov (KS) test, which can be used to compare the distribution of the new data set $\{dist_{test}^{c_k}\}$ to that of the original data set $\{dist_{train}^{c_k}\}$. If the two distributions are found to be different, it can be inferred that the two data sets come from different distributions, thereby indicating the presence of a drift in data set $\{s_i^{c_k}\}$.

If a neural network has l layers, with $g_1, g_2, \ldots, g_l$ neurons in each layer, the vector representation of each layer's outputs, denoted as $Z^1, Z^2, \ldots, Z^l$, will have the dimensionality $g_1, g_2, \ldots, g_l$. The dimension of this representation is equal to the number of neurons in the layer Z^L and is typically quite large. The nodes z_i^L in the hidden layer Z^L of the network can be calculated using the equation $z_i^L = f(w \cdot x + b)$, where w represents the weights, b is the bias vector, and f is the activation function. Neural network's hierarchical learning architecture allows them to automatically extract representations of varying complexities from the input data. We choose a vector representation of some selected hidden layer to represent the original input data. In comparison to the original input data, activation values are a more meaningfully extracted representation, and testing for concept drift on these values can make the detection process more efficient. Our detection method can be applied to an already trained neural network model during the reference stage, when the model is being used to process new data. Our method chooses one of the layer's activation values as the inputs for the drift detection process. If a concept drift occurs, the outputs of the medium layers in the network will also reflect the drift. Activation values of different layers provide different levels of representation of the data, and using our method on the outputs of different layers can help us to identify different sources of concept drift, which can inform the development of appropriate remedial measures. The output layer is typically more constrained in structure than the hidden layers, and typically represents the final classification results. This layer is less likely to contain many of the original features present in the input data.

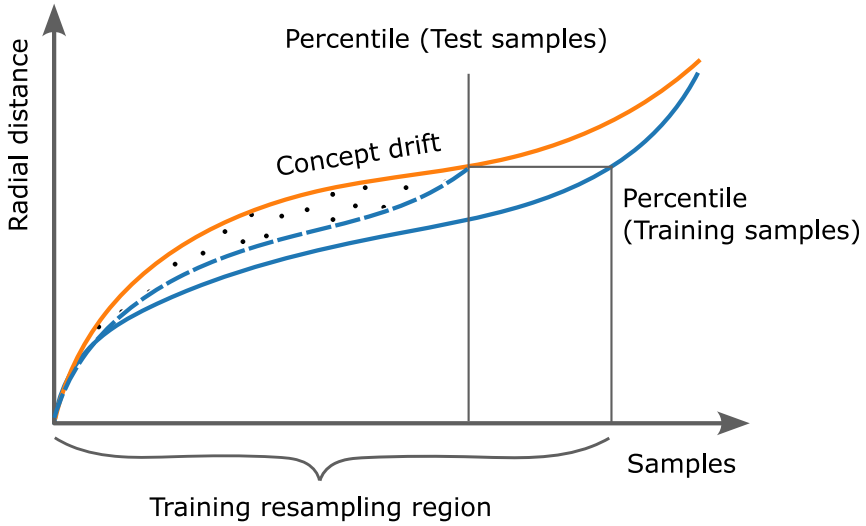
The activation function is a crucial component of the calculation and is typically chosen from among several commonly used functions, such as the hyperbolic tangent (Tanh) function, the logistic function, or the rectified linear unit (Relu) function. These activation functions are characterized by their S-shaped curve, which maps the real-valued input to a small value range, such as between zero and one. The outputs of each neuron tend to cluster around these values, potentially simulating the behavior of biological neurons. In classification settings, the nodes in each layer of the neural network possess two important properties. Firstly, the activated neurons for samples of the same category are similar, and secondly, the activated neurons for samples of different categories are different. We leverage these two properties by detecting drifts on each class cluster. For each class cluster, we can choose the radial base based on the dimensions with higher values $B(Z^L, \gamma) = \left\langle \begin{cases} (\lambda * \overline{Z_j^L}) & \text{if } \left(\overline{Z_j^L}\right) > percentile(\overline{Z^L}, \gamma) \\ \overline{Z_i^L} & \text{otherwise} \end{cases} \right\rangle$($\lambda$ is the integral multiplier of $\overline{Z_j^L(x)}$, $\overline{Z_j^L(x)}$ is the mean activation value of neuron j given all x_{train}), as these are more likely to provide useful information for that class. In particular, for a specific class of data cluster, some neurons may have values that are higher, while others may be lower and near zero. By considering only the dimensions with higher values, we can increase the accuracy and efficiency of the drift detection method. By adjusting the location of the radial distance base, we can effectively tune the sensitivity of detection methods on different features. This is particularly useful when applied to neural network models, because the activated neurons for samples of different categories are different.

When comparing two batches of one-dimensional radial distance data sets in the drift detection phase, our method utilizes the KS test instead of a permutation test. This further reduces the computation cost, while still providing robust results in many cases. It is important to note that our method may not perform as well in situations where changes to unrelated features for that class cluster, as it introduce a bias that affects the accuracy of the results. But changes in related features are more prevalent, as is often the case in real-world applications. In addition to its low computation cost, another advantage of our method is that it reduces the data storage requirements by using radial distances as the sole feature representation for each data point. Unlike traditional multidimensional two-sample test methods, our method does not require the restoration of the entire high-dimensional data set. Instead, we only need to compute and store the radial distances. This results in a significant reduction in memory requirements. The complexity of the proposed algorithm is $O(n)$ for computing high-dimensional distance from each sample point to the radial base and $O((n+m)\log(n+m))$ for computing the KS-test statistics.

In order to distinguish novelty from concept drift, we further examine difference in the radial base distances $\{dist^{c_k}\}$ for these two types of distribution changes. Novelty samples reflected in radial base distances present higher values. We leverage this property and selecting a percentile value $d(P)$ = percentile($dist_{test}^{c_k}$,P), (P is the re-sampling percentile), such that higher values of radial distances D'_{test}=$\{dist_{test}^{c_k}, dist_{test}^{c_k} > d(P)\}$, which correspond to the novelty samples can be excluded. The remaining radial distances of the test samples D_{test}=$\{dist_{test}^{c_k}, dist^{c_k} < d(P)\}$should have same distribution as those of the training samples D_{train}=$\{dist_{train}^{c_k}, dist_{train}^{c_k} < d(P1)\}$((P1 is the training re-sampling percentile)). Thus, we first apply KS test on remaining test samples D_{test} against training samples D_{train}, Then we apply KS test again on the full samples$\{dist_{train}^{c_k}\}$, $\{dist_{test}^{c_k}\}$. If the first test KS(D_{test}, D_{train}) indicates same distribution but second test KS($\{dist_{train}^{c_k}\}$, $\{dist_{test}^{c_k}\}$)does not, that means the test data has novelty distribution change. On the other hand, if both tests indicate different distributions, that means the test data has concept drift distribution change.



(a) Radial base distances on training data and test data with novelty distribution change.



(b) Radial base distances on training data and test data with concept drift distribution change.

Fig. 3. Concept drift and novelty distribution changes reflected on radial base distances between training data and test data. Sub-figure (a) illustrates the partial higher values of radial distances caused by the novelty distribution change from the test data (marked in orange curve). Given a percentile to exclude these high values, the remaining radial distances of the test data and the training data present same distributions. Sub-figure (b) illustrates the overall higher values of radial distances caused by the concept drift distribution change from the test data (marked in orange curve). Given a percentile to exclude the high values, the remaining radial distances of the test data and the training data still present significantly different distributions.

Fig. 3 illustrates the difference between concept drift and novelty distribution changes, reflected on radial base distances between training data and test data. Sub-figure (a) illustrates the partial higher values of radial distances caused by the novelty distribution change from the test data (marked in orange curve). Given a percentile to exclude these high values, the remaining radial distances of the test data and the training data present same distributions. Sub-figure (b) illustrates the overall higher values of radial distances caused by the concept drift distribution change from the test data (marked in orange curve). Given a percentile to exclude the high values, the remaining radial distances of the test data and the training data still present significantly different distributions.

Algorithm 2 provide our novelty-aware concept drift detection method using activation outputs of neural network's hidden layer.

4. Experimental evaluation

In this section, we evaluate the proposed novelty-aware concept drift detection method with eight experiments, grouped into four sections. Experiment 1 visualizes the proposed statistic — radial based distance, demonstrating differences between concept drift and novelty distribution changes. Experiments 2–5 are designed to show the distribution change detection accuracy of the proposed method and the impact of various data dimensions and sample sizes on the accuracy and

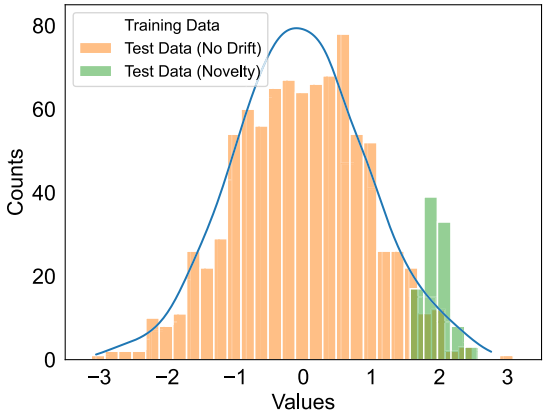
Algorithm 2: Novelty-aware Concept Drift Detection (NACD)

input : X, new input data samples from test dataset or stream, predicted as a specific class;
Z^L, activation of layer L in neural network model;
P, re-sampling percentile;
B, radial distance base;
$Dist$, distance function, by default Euclidean;
R, referential radial distances;
T, mutable list of testing radial distances, by default empty list;
θ, confidence threshold for drift detection, by default 0.05.

output: boolean concept drift detection result.
boolean novelty detection result.

get activation $Z^L(X)$ for input X;
compute $T = Dist(Z^L(X), B)$, the distance from activation vector to base vector;
append $Dist(Z^L(x), B)$ to the end of T;
get the test percentile value $d(P)$ according to percentile P;
re-sample test samples T_p according to P;
re-sample retraining samples R_p according to $P1$;
apply Kolmogorov–Smirnov test to R_p and T_p and get P-value $KS(R_p, T_p)$;
if $KS(R_p, T_p) < \theta$ **then**
    return *True (drift), False (no novelty)*;
else
    apply Kolmogorov–Smirnov test to R and T and get P-value $KS(R, T)$;
    if $KS(R, T) < \theta$ **then**
        return *False (no drift), True (novelty)*;
    else
        return *False (no drift), False (no novelty)*

(a) Radial base distances on concept drift data



(b) Radial base distances on novelty data

Fig. 4. Two sample sets generated in Gaussian distribution, with different parameters to simulate radial base distances computed from data samples with concept drift and novelty distribution changes between training data and test data. Sub-figure (a) shows the distribution difference between training data (marked in blue line) and test data (marked in orange bar) with concept drift distribution change. The training data is generated in Gaussian distribution with mean 0, and the test data with mean 0.5. They have same variance 1.0. Sub-figure (b) shows the distribution difference between training data (marked in blue line) and test data (marked in orange and green bars) with novelty distribution change. The training data is generated in Gaussian distribution with mean 0, variance 1.0. The test data consists of two parts. The majority of the test samples are still generated with mean 0.5 and variance 1.0. A small part of test samples are generated with mean 0.2, variance 0.1. This is to simulate the high values of the radial base distances caused by the novelty samples.

efficiency. Experiments 6–7 apply our method to neural networks with real-world image classification data sets, to verify its detection accuracy for both concept drift and novelty. Finally, experiment 8 demonstrates the impact of different choices of algorithm parameters on the detection accuracy.

4.1. Novelty-aware radial base distance

Our method is based on the radial base distance statistic, to measure the distribution changes of neural network layer activation. Radial base distance not only serves as the foundation of the proposed detection method, the statistic itself is a general tool with the potential for developing new test methods. Thus, we first design experiment to further demonstrate the behavior of the statistic under different kinds of data distribution changes.

Experiment 1. Algorithm visualization

In this experiment, we generate two sample sets in Gaussian distribution, with different parameters to simulate radial base distances computed from data samples with concept drift and novelty distribution changes between training data and test data. As illustrated in Fig. 4, sub-figure (a) shows the distribution difference between training data (marked in blue line) and test data (marked in orange bar) with concept drift distribution change. The training data is generated in Gaussian distribution with mean 0, and the test data with mean 0.5. They have same variance 1.0. Sub-figure (b) shows the distribution difference between training data (marked in blue line) and test data (marked in orange and green bars) with novelty distribution change. The training data is generated in Gaussian distribution with mean 0, variance 1.0. The test data consists of two parts. The majority of the test samples are still generated with mean 0.5 and variance 1.0. A small part of test samples are generated with mean 0.2, variance 0.1. This is to simulate the high values of the radial base distances caused by the novelty samples.

In the proposed algorithm, we sort the radial base distances in ascending order for training data, test data with concept drift and test data with novelty respectively. As illustrated in Fig. 5, the radial base distances of the training data (marked in blue line) present a smooth curve, which is considered as the reference. The radial base distances of the test data with concept drift (marked in orange line) show overall higher values, but remain the smooth curve. The radial base distances of test data with novelty (marked in green line) first show a smooth curve in low value range, then a noticeable increase in the high value range, which is caused by the novelty samples. With percentile set to 800 (out of total sample size 1000), P1 and P2 are the radial base distance values corresponding to the percentile for both concept drift

Table 1
Additional parameters of comparison methods used in the experiments.

Method	Parameter	Symbol	Value
Ours	Percentile threshold	σ	0.8
CM	Euclidean distance threshold	d_ε	0.05
MMD	Kernel bandwidth	α	0.01
KDQ	Maximum number of points in a cell	τ	20
	Minimum side length of a cell	δ	0.01
LDD	Neighborhood ratio	ρ	0.1
	Drift significance level	α	0.05

data and novelty data. R1 and R2 are the referential percentile on the radial base distances of the training data corresponding to P1 and P2 respectively.

Finally, we re-sampled radial base distances for the training data according to the referential percentiles, and compare them with the percentile samples of the radial base distances of the test data with concept drift and novelty. As illustrated in Fig. 6, sub-figure (a) shows that the re-sampled radial base distances of the test data with concept drift (marked in orange line) overall have higher values than the re-sampled radial base distances of the training data (marked in blue line). Sub-figure (b) shows that the re-sampled radial base distances of the test data with novelty (marked in green line), have the almost same distribution as the re-sampled radial base distances of the training data (marked in blue line).

This experiment shows that the re-sampled radial base distances for both training data and test data will have different distribution for concept drift but same distribution for novelty. This property of the radial base distances is the foundation of the proposed novelty-aware concept drift detection method.

4.2. Drift and novelty detection

In this section, we design experiments with synthetic data to first show the detection accuracy of the proposed method for both concept drift and novelty, and then extend the data to various dimensions and sample sizes to show their impact on its accuracy and efficiency.

For comparison, we choose four representative works of distribution change detection method — competence model-based drift detection (CM) [27], Maximum mean discrepancy(MMD) [6], local drift degree-based drifted instance selection algorithm(LDD) [44], and kdq-tree based change detection method(KDQ) [45]. Unless indicated otherwise, the training data and test data are generated with multivariate Gaussian distribution with 15 dimensions, mean 1.0 in each dimension and correlation 1.0 between dimensions. The window size for distribution tests is set to 200; the significance level is set to 5% for all the algorithms. Algorithm specific parameters are listed in Table 1. We repeat the full process 10 times, and report the mean detection accuracy.

Experiment 2. Detection accuracy

In this experiment, we evaluate the performance of the proposed detection method against the various magnitude of distribution changes. First, for the test data, we introduce an offset ranging in (0.004, 0.005, 0.006, 0.007) to simulate concept drift. A total of 100 batches of training data and test data are generated, and we apply our method, CM, MMD, KDQ and LDD to the batches to compare their concept drift detection accuracy. As shown in Fig. 7(a), the detection accuracy of all five methods increase as the offset of the drift increases. However, our method outperforms other methods by a large margin in all cases. This indicates that our method has higher sensitivity on concept drift distribution changes.

Next, this time for the test data, we apply an offset ranging in (0.02, 0.025, 0.03, 0.035) to a small portion of the samples (20%) to simulate novelty samples. A total of 100 batches of training data and test data are generated, and we apply our method, CM, MMD, KDQ and LDD to the batches to compare their novelty detection accuracy. As shown in Fig. 7(b), the detection accuracy of all five methods increase as the offset of the novelty increases. However, our method outperforms other methods in four cases and ranks the second in the remaining one case. This indicates that our method has relatively high sensitivity on novelty distribution changes.

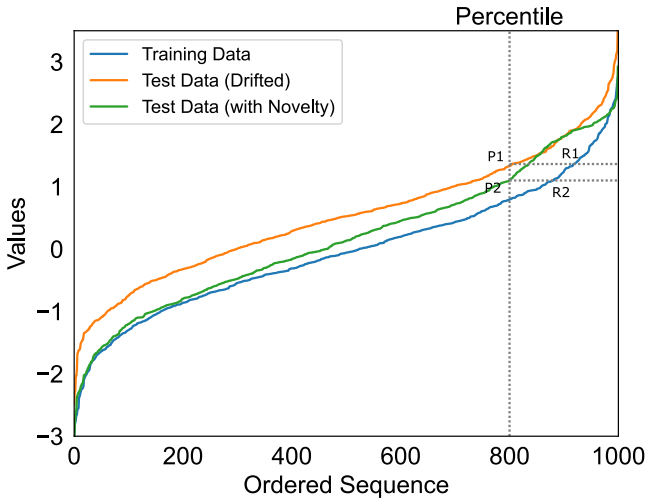
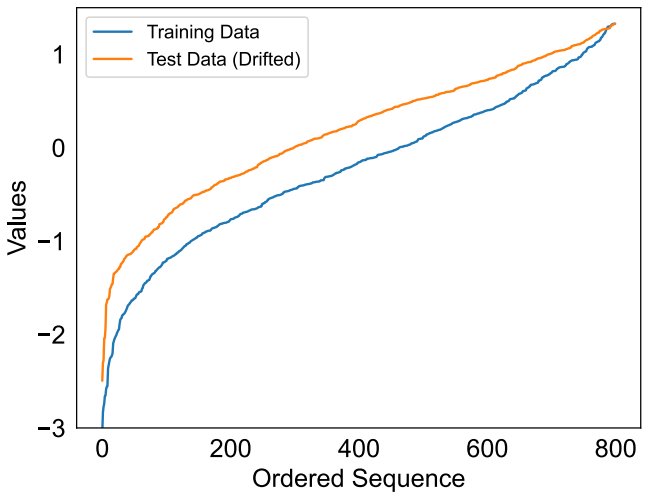


Fig. 5. Radial base distances sorted in ascending order for training data, test data with concept drift and test data with novelty respectively. The radial base distances of the training data (marked in blue line) present a smooth curve, which is considered as the reference. The radial base distances of the test data with concept drift (marked in orange line) show overall higher values, but remain the smooth curve. The radial base distances of test data with novelty (marked in green line) first show a smooth curve in low value range, then a noticeable increase in the high value range, which is caused by the novelty samples. With percentile set to 800 (out of total sample size 1000), P1 and P2 are the radial base distance values corresponding to the percentile for both concept drift data and novelty data. R1 and R2 are the referential percentile on the radial base distances of the training data corresponding to P1 and P2 respectively.

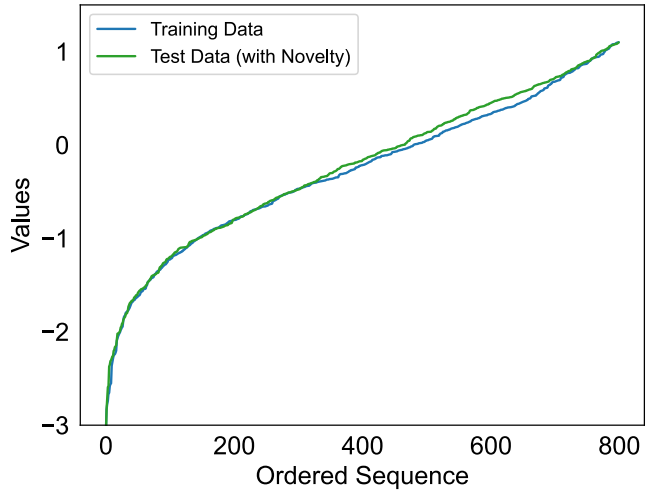
Experiment 3. High dimension

In this experiment, we evaluate the impact of various dimensions of the data on the detection accuracy of proposed method. First, for the test data, we introduce an offset 0.007 to simulate concept drift. However, for different batches, the offset is applied to different dimensions varying in range 3–7. A total of 100 batches of training data and test data are generated, and we apply our method, CM, MMD, KDQ and LDD to the batches to compare their concept drift detection accuracy. As shown in Fig. 8(a), the detection accuracy of all five methods increase as the offset is applied to more dimensions. However, our method outperforms other methods in most cases. With similar settings, we fix the number of drift dimensions to 7, and vary the number of total dimensions of the data in range 20–70. As shown in Fig. 8(b), the detection accuracy of all five methods remains unchanged as total dimension increases. However, our method outperforms other methods in all cases. This indicates that our method has higher sensitivity on concept drift distribution changes in higher dimensions.

Next, for the test data, we introduce an offset 0.035 to a small part of the test samples (20%) to simulate novelty. However, for different batches, the offset is applied to different dimensions varying in range 3–7. A total of 100 batches of training data and test data are generated, and we apply our method, CM, MMD, KDQ and LDD to the batches to compare their concept drift detection accuracy. As shown in Fig. 9(a), the detection accuracy of all five methods increase as the offset is applied to more dimensions. Our method has similar performance as CM but lower than MMD. With similar settings, we fix the number of drift dimensions to 7, and vary the number of total dimensions of the data in range 20–70. As shown in Fig. 9(b), the detection accuracy of all three methods remains unchanged as total dimension increases. Our method ranks the second in most cases and lower than MMD.

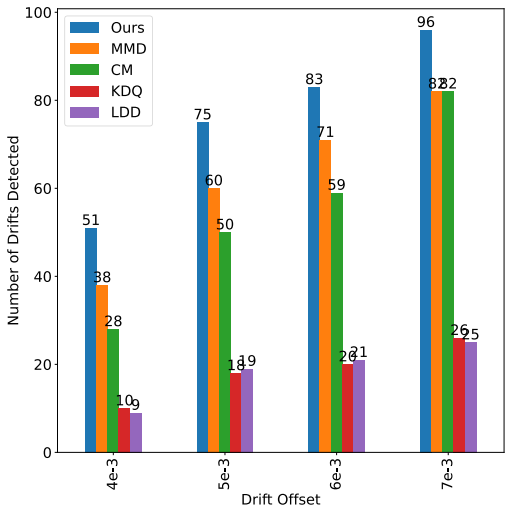


(a) Re-sampled radial base distances for concept drift

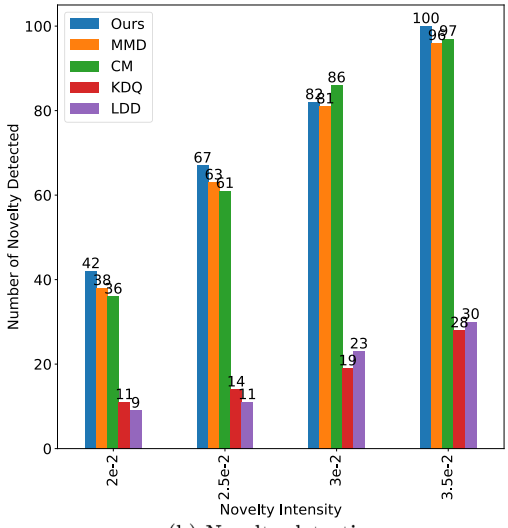


(b) Re-sampled radial base distances for novelty

Fig. 6. Radial base distances re-sampled for the training data according to the referential percentiles, and compared with the percentile samples of the radial base distances of the test data with concept drift and novelty. Sub-figure (a) shows that the re-sampled radial base distances of the test data with concept drift (marked in orange line) overall have higher values than the re-sampled radial base distances of the training data (marked in blue line). Sub-figure (b) shows that the re-sampled radial base distances of the test data with novelty (marked in green line), have the almost same distribution as the re-sampled radial base distances of the training data (marked in blue line).



(a) Concept drift detection



(b) Novelty detection

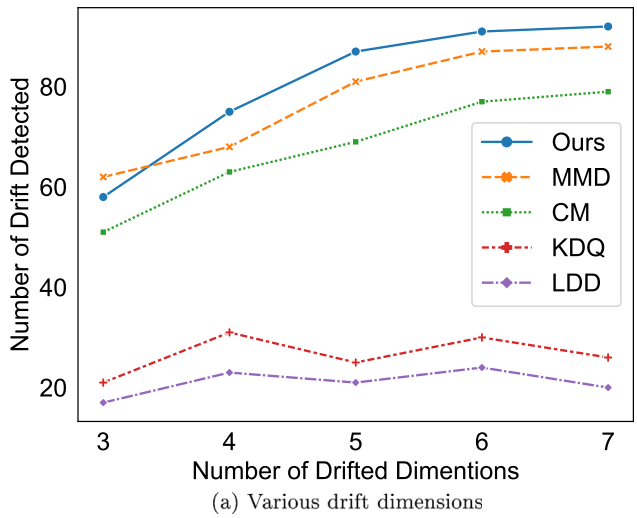
Fig. 7. Concept drift and novelty distribution change detection accuracy against the various magnitude of distribution changes. For the test data, in sub-figure (a), an offset ranging in (0.004, 0.005, 0.006, 0.007) is introduced to simulate concept drift; in sub-figure (b), an offset ranging in (0.02, 0.025, 0.03, 0.035) is applied to a small portion of the samples (20%) to simulate novelty samples. A total of 100 batches of training data and test data are generated, and we apply our method, CM, MMD, KDQ and LDD to the batches to compare their concept drift detection accuracy.

This shows that our method has comparative sensitivity on novelty distribution changes in different dimensions.

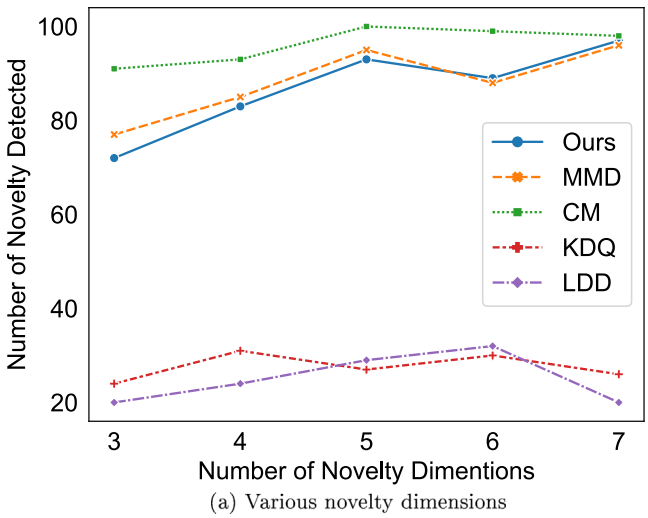
Experiment 4. Window size

In this experiment, we evaluate the impact of various window sizes of the data on the detection accuracy of proposed method. First, for the test data, we introduce an offset 0.007 to simulate concept drift. For different batches, the window sizes of training data and test data vary in range (100, 150, 200, 250). A total of 100 batches of training data and test data are generated, and we apply our method, CM, MMD, KDQ and LDD to the batches to compare their concept drift detection accuracy. As shown in Fig. 10(a), the detection accuracy of all three methods increase as the window size increases. However, our method outperforms other methods in most cases. This indicates that our method has higher sensitivity on concept drift distribution changes with different window sizes.

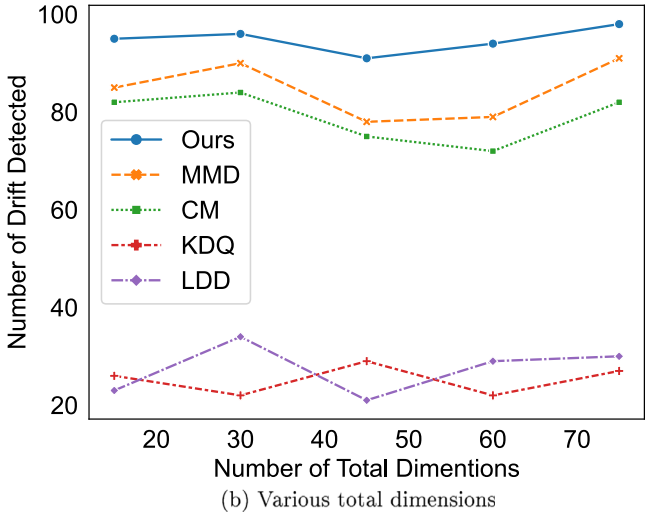
Next, for the test data, we introduce an offset 0.035 to a small part of the test samples (20%) to simulate novelty. For different batches, the window sizes of training data and test data vary in range (100, 150, 200, 250). A total of 100 batches of training data and test data are generated, and we apply our method, CM, MMD, KDQ and LDD to the batches to compare their concept drift detection accuracy. As shown in Fig. 10(b), the detection accuracy of all five methods increase as the window size increases. However, our method outperforms other methods in all cases. This shows that our method has comparative sensitivity on novelty distribution changes with different window sizes.



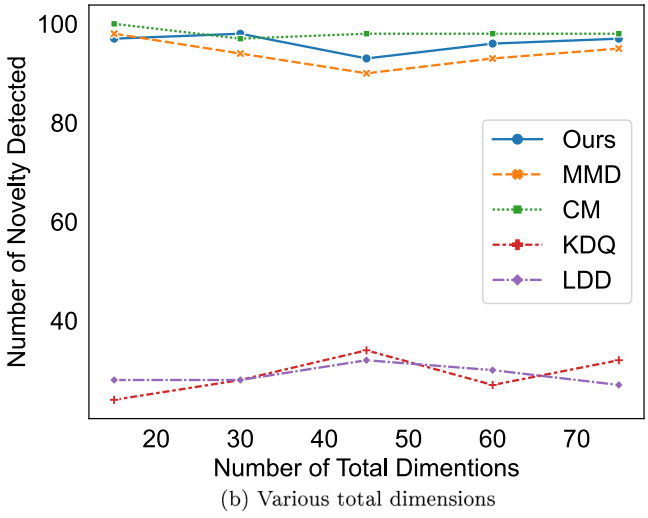
(a) Various drift dimensions



(a) Various novelty dimensions



(b) Various total dimensions



(b) Various total dimensions

Fig. 8. Detection accuracy of concept drift distribution changes in different dimensions. A total of 100 batches of training data and test data are generated. In sub-figure (a), fix the number of total dimensions to 15 and vary the number of drift dimensions in range 3–7. In sub-figure (b), With similar settings, fix the number of drift dimensions to 7, and vary the number of total dimensions of the data in range 20–70.

Fig. 9. Detection accuracy of novelty distribution changes in different dimensions. A total of 100 batches of training data and test data are generated. In sub-figure (a), fix the number of total dimensions to 15 and vary the number of novelty dimensions in range 3–7. In sub-figure (b), With similar settings, fix the number of novelty dimensions to 7, and vary the number of total dimensions of the data in range 20–70.

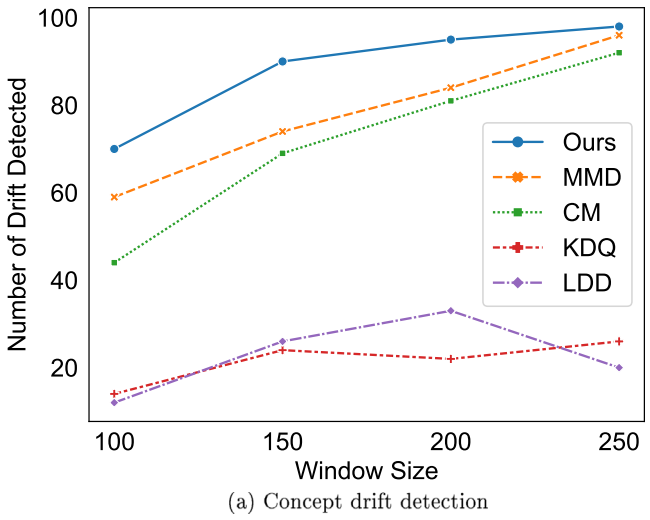
Experiment 5. Efficiency

Our method requires a one-time preprocessing step to transform the input data into a suitable one-dimensional value. This step involves calculating the distance from the radial base, which has a time complexity of $O(n)$, where n is the number of data points. In the K-S test stage, the complexity is $O((n+m)\log(n+m))$. The computational complexity of MMD is at least $O(n^2)$. The CM method also have a time complexity of $O(n^2)$ and Permutation Test in CM method adds an additional factor of N to the complexity, resulting in $O(N \times n^2)$. N is the number of permutations. In this experiment, we measure the computation time cost of proposed method with data sets of various dimensions and window sizes and compare it with other methods. The running times are obtained in a server environment with Intel Xeon 2.60 GHz CPU, 256 GB memory and 64 bit Red Hat Linux Operating System. The programs are implemented in Python 3.9 with numpy, scipy library stack. No parallel computation is used for easier performance analysis.

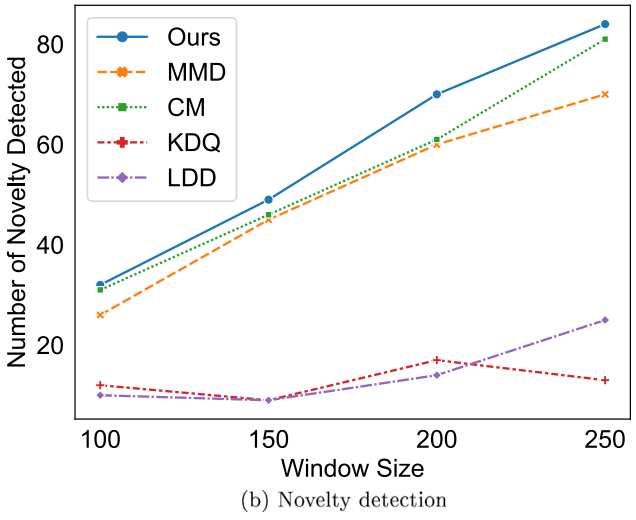
The result is shown in Fig. 11. The key observations is that, the detection process of our method is very efficient, outperforms other methods by approximately two magnitudes. Sub-figure (a) shows the computation time of detection methods with data of different number of dimensions, ranging in 20–70. Sub-figure (b) shows the computation time of detection methods with data of different window sizes, ranging in 100–250. Our method is about 100 times faster than other methods. As the number of dimensions increase, the computation time of all three method increases linearly. Noticeably, comparing to our method and other methods computation time increases faster as window size increases. LDD consumes the most computation time.

4.3. Real-world data

In this section, we establish experiments to apply the proposed method to neural networks on real-world image classification tasks, to verify its detection accuracy for both concept drift and novelty. We use the **CIFAR-10** data set [46], which is a image classification benchmark data set. The data set consists of 60 000 32 × 32 color images in 10 classes, with 6000 images per class. There are 50 000 training images and 10 000 test images. The data set is divided into five training batches and one test batch, each with 10 000 images. The test

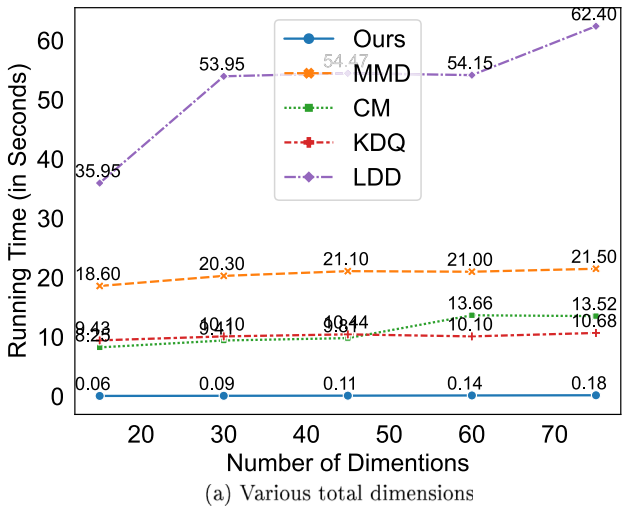


(a) Concept drift detection

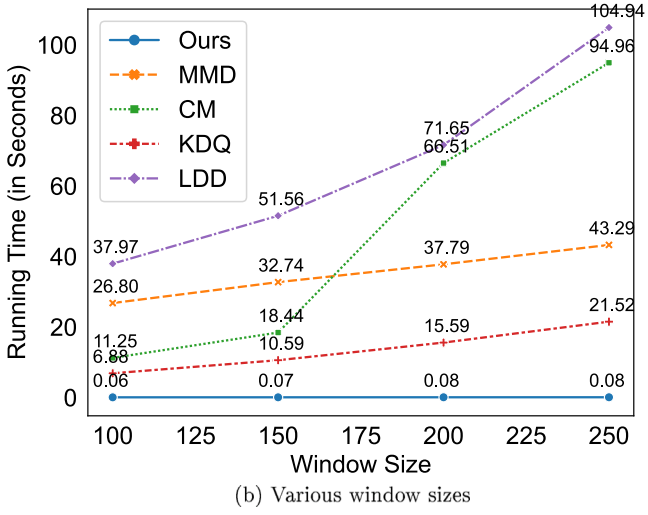


(b) Novelty detection

Fig. 10. Detection accuracy of concept drift detection and novelty detection on data with different window sizes in range 100–250. A total of 100 batches of training data and test data are generated. Sub-figure (a) shows the number of detected concept drift out of 100. Sub-figure (b) shows the number of detected novelty.



(a) Various total dimensions



(b) Various window sizes

Fig. 11. Computation time of the detection methods on data with concept drift and novelty. Lower values means higher computation efficiency. Sub-figure (a) shows the computation time of detection methods with data of different number of dimensions, ranging in 20–70. Sub-figure (b) shows the computation time of detection methods with data of different window sizes, ranging in 100–250. Our method is about 100 times faster than other methods.

batch contains exactly 1000 randomly-selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another. The classes are completely mutually exclusive. The base neural network used for the detection methods has a typical convolutional neural network structure, as shown in Table 2.

Experiment 6. Concept drift detection

In this experiment, we evaluate the performance of the proposed detection method with real-world image classification data with concept drift. As shown in Fig. 12(a), non-modified CIFAR-10 images are used as training data for the neural network. Then for the test data, as shown in Fig. 12(b), Gaussian blurring is applied to the CIFAR-10 images to simulate concept drift.

We apply our method, CM, MMD, KDQ and LDD to two randomly selected layers of the base neural network, namely F1 and F3 as listed in Table 2, to detect the concept drift in the test data. Fig. 13(a) shows the concept drift detection result on neural network layer F1 for different concept drift intensity, which is simulated by varying the kernel bandwidth parameter of the Gaussian blur within range 0.46–0.47. Fig. 13(b) shows the concept drift detection result on neural network layer F3 for different concept drift intensity, which is simulated by varying the kernel bandwidth parameter of the Gaussian blur within range 0.55–0.60. The results show that the proposed method achieves relatively high accuracy comparing to other methods. KDQ and CM detect less drifts than other three methods on layer F1. MMD detects less drifts than other four methods on layer F3.

Experiment 7. Novelty distribution change detection

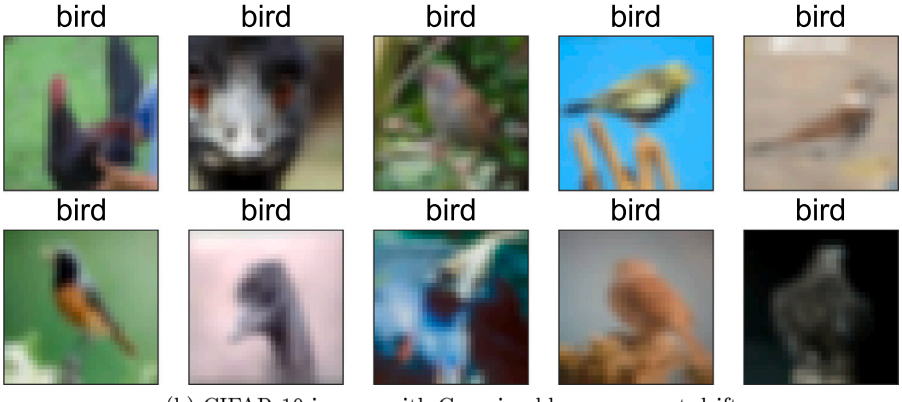
In this experiment, we evaluate the performance of the proposed detection method with real-world image classification data with novelty. As shown in Fig. 14(a), non-modified CIFAR-10 images are used as training data for the neural network. Then for the test data, as shown in Fig. 14(b), new image categories that was not used for training is added to test data to simulate novelty distribution change.

We apply our method, CM, MMD, KDQ and LDD to two randomly selected layers of the base neural network, namely F1 and F3 as listed in Table 2, to detect the novelty distribution change in the test

Table 2
Convolutional neural network structure used as the base neural network for concept drift and novelty distribution change detection.

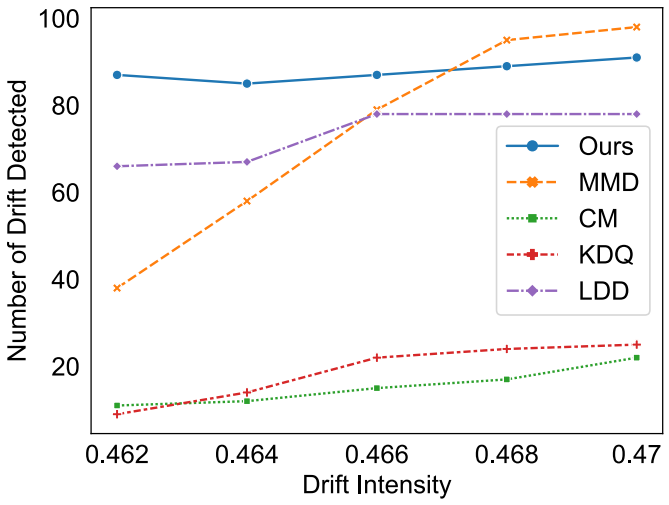
Layer ID	Layer type	Parameters
C0	Conv2d	size 3 × 32, kernel size 3 × 3, stride 1 × 1, padding 1 × 1
C1	BatchNorm2d	size 32, eps 1e−05, momentum 0.1
C2	ReLU	
C3	Conv2d	size 32 × 64, kernel size 3 × 3, stride 1 × 1, padding 1 × 1
C4	ReLU	
C5	MaxPool2d	kernel size 2, stride 2, padding 0, dilation 1
C6	Conv2d	size 64 × 128, kernel size 3 × 3, stride 1 × 1, padding 1 × 1
C7	BatchNorm2d	size 128, eps 1e−05, momentum 0.1
C8	ReLU	
C9	Conv2d	size 128 × 128, kernel size 3 × 3, stride 1 × 1, padding 1 × 1
C10	ReLU	
C11	MaxPool2d	kernel size 2, stride 2, padding 0, dilation 1
C12	Dropout2d	probability 0.05
C13	Conv2d	size 128 × 256, kernel size 3 × 3, stride 1 × 1, padding 1 × 1
C14	BatchNorm2d	size 256, eps 1e−05, momentum 0.1
C15	ReLU	
C16	Conv2d	size 256 × 256, kernel size 3 × 3, stride 1 × 1, padding 1 × 1
C17	ReLU	
C18	MaxPool2d	kernel size 2, stride 2, padding 0, dilation 1
F0	Dropout2d	probability 0.1
F1	Linear	in features 4096, out features 1024
F2	ReLU	
F3	Linear	in features 1024, out features 512
F4	ReLU	
F5	Dropout2d	probability 0.1
F6	Linear	in features 512, out features 10

(a) CIFAR-10 original images

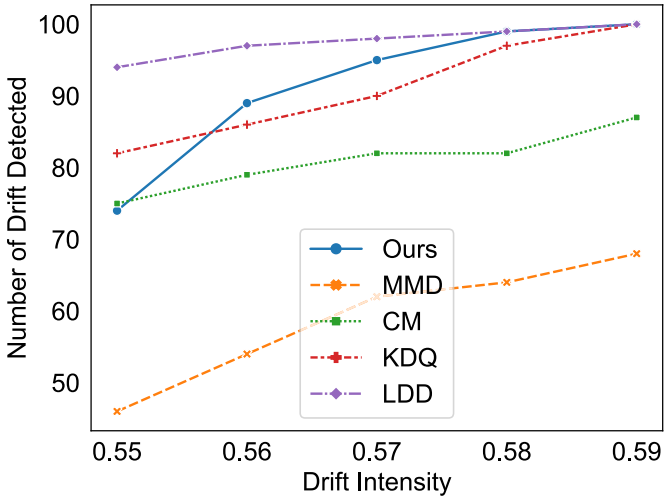


(b) CIFAR-10 images with Gaussian blur as concept drift

Fig. 12. CIFAR-10 data set is used for neural network training and concept drift detection. In sub-figure (a), non-modified CIFAR-10 images are used as training data for the neural network. Then for the test data, as shown in sub-figure (b), Gaussian blurring is applied to simulate concept drift.



(a) Concept drift detection on network layer F1



(b) Concept drift detection on network layer F3

Fig. 13. Concept drift detection result of our method, MMD and CM on two randomly selected layers of the base neural network, namely F1 and F3 as listed in Table 2. Sub-figure (a) shows the concept drift detection result on neural network layer F1 for different concept drift intensity, which is simulated by varying the kernel bandwidth parameter of the Gaussian blur within range 0.46–0.47. Sub-figure (b) shows the concept drift detection result on neural network layer F3 for different concept drift intensity, which is simulated by varying the kernel bandwidth parameter of the Gaussian blur within range 0.55–0.60.

data. Fig. 15(a) shows the novelty concept drift detection result on neural network layer F1 for different number of novelty samples added, within range 20–40. Fig. 15(b) shows the novelty detection result on neural network layer F3 with similar settings. The results show that the proposed method achieves relatively high accuracy comparing to other methods. Noticeably, the detection accuracy of our method and MMD increases faster than other three methods as the number of novelty samples increases.

Parameter analysis

In this part, we demonstrates the impact of different choices of algorithm parameters on the detection accuracy.

Experiment 8. The impact of percentile and multiplier on detection accuracy

There are two parameters in out method: Percentile (resampling percentile in novelty distribution change detection) and Multiplier (integral multiplier of $\overline{Z_j^L(x)}$ in radial base). Both concept drift and novelty distribution change are evaluated using the CIFAR-10 image data set. Gaussian blurring of kernel bandwidth 0.56 is applied to simulate concept drift; 20% of images from new class are added to simulate novelty distribution change. The result is shown in Fig. 16. We can see that for concept drift detection, (1) larger Percentile leads to higher accuracy, which is expected since more samples are used for the detection; (2) small Multiplier greatly impedes detection accuracy, yet values larger than 4 have no difference and yield similar result. Contrastingly, for novel distribution change detection, (1) best result is acquired using a Percentile lower than but close to the proportion of the novel samples, which is expected since this maximizes the number of novel samples used for distinguishing them; (2) Multiplier between 2 and 4 yields best result, since the novel samples only increases a part of the radial base distances, while overly large multipliers increase the overall distances which dominate the partial distance distribution change. Thus, the recommended multiplier is 2 to 5; the percentile should be chosen according to the estimated proportion of possible novel samples.

5. Conclusion and further study

In conclusion, the study focuses on the challenges faced by neural network models in processing streaming data. The research community has attempted to address these challenges, but existing methods are unable to distinguish between concept drift and novelty, leading to inappropriate allocation of model maintenance resources. The proposed concept drift detection method is novel and capable of distinguishing between the two, making it a promising solution to the problem of accuracy degradation in real-world applications. Additionally, the method is more efficient than existing drift detection methods, making it suitable for use in large-scale neural network applications. Our next goal is to develop adaptive model maintenance algorithms based on the statistics output by our new concept drift detection method results. An alternative research direction could be to extending the algorithm to regression models and other non-classification tasks.

CRediT authorship contribution statement

Dan Shang: Writing – original draft. **Guangquan Zhang:** Supervision. **Jie Lu:** Supervision.

Declaration of competing interest

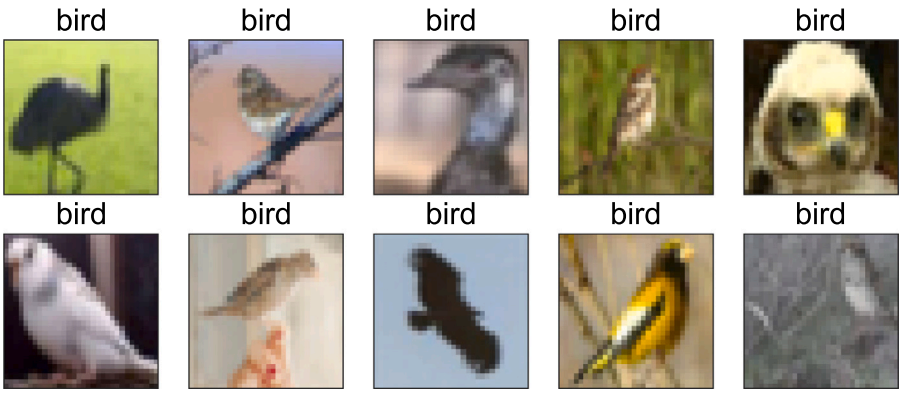
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The work presented in this paper was supported by the Australian Research Council (ARC) under Discovery Project DP220102635.

Data availability

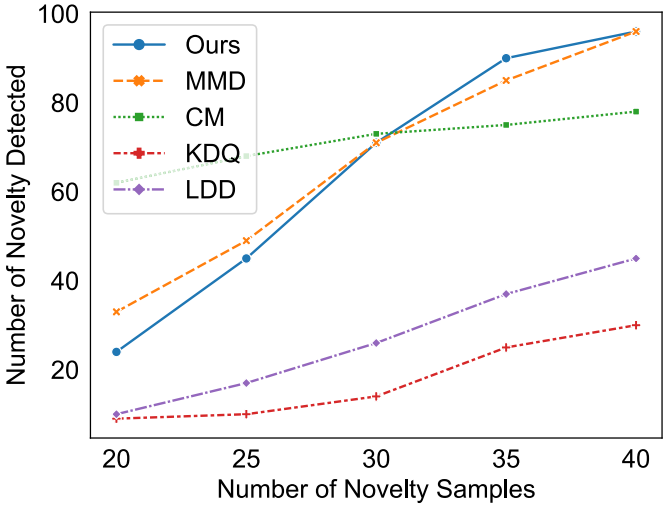
Data will be made available on request.



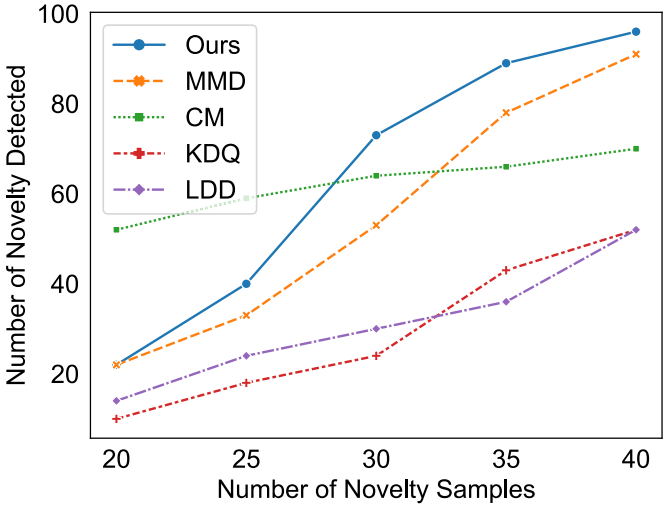
(a) CIFAR-10 original images

(b) CIFAR-10 images not exist in training data as novelty

Fig. 14. CIFAR-10 data set is used for neural network training and novelty detection. In sub-figure (a), non-modified CIFAR-10 images are used as training data for the neural network. Then for the test data, as shown in sub-figure (b), new image categories that was not used for training is added to simulate novelty distribution change.

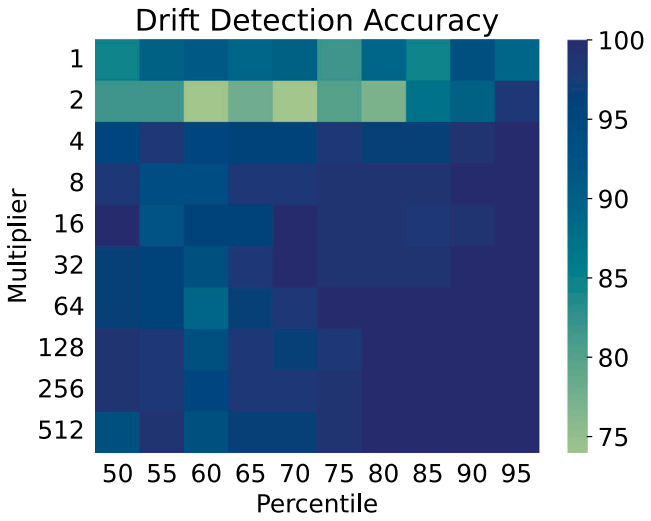


(a) Novelty distribution change detection on network layer F1

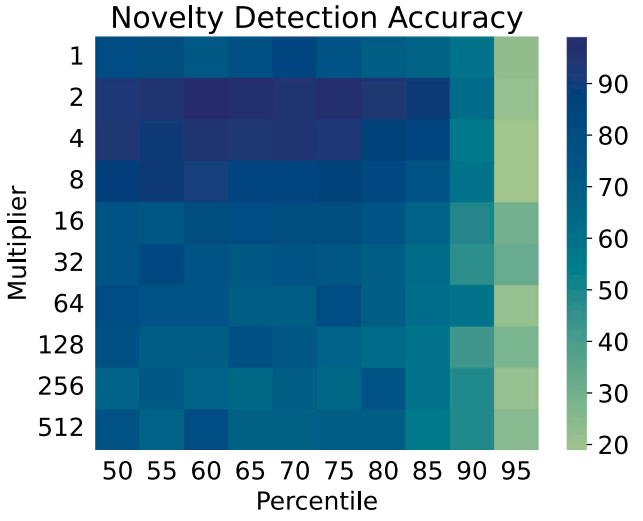


(b) Novelty distribution change detection on network layer F3

Fig. 15. Apply our method, MMD and CM to two randomly selected layers of the base neural network, namely F1 and F3 as listed in Table 2, to detect the novelty distribution change in the test data. Sub-figure (a) shows the novelty detection result on neural network layer F1 for different number of novelty samples added, within range 20–40. Sub-figure (b) shows the novelty distribution change detection result on neural network layer F3, with similar settings.



(a) Concept drift detection accuracy. Darker blocks indicate higher accuracy



(b) Novelty detection accuracy. Darker blocks indicate higher accuracy

Fig. 16. Impact on detection accuracy using different algorithm parameters. Radial Base multiplier increases exponentially from 1 to 512; percentile increases from 50 to 95.

References

[1] Heng Wang, Zubin Abraham, Concept drift detection for streaming data, in: 2015 International Joint Conference on Neural Networks (IJCNN), 2015.

[2] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, Abdelhamid Bouchachia, A survey on concept drift adaptation, ACM Comput. Surv. 46 (4) (2014) 1–37.

[3] Yoshiaki Yasumura, Naho Kitani, Kuniaki Uehara, Quick adaptation to changing concepts by sensitive detection, in: Hiroshi G. Okuno, Moonis Ali (Eds.), International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems - IEA/AIE 2007, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 855–864.

[4] Peipei Li, Xuegang Hu, Qianhui Liang, Yunjun Gao, Concept drifting detection on noisy streaming data in random ensemble decision trees, in: Petra Perner (Ed.). International Workshop on Machine Learning and Data Mining in Pattern Recognition - MLDM 2009, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 236–250.

[5] Norbert Henze, A multivariate two-sample test based on the number of nearest neighbor type coincidences, Ann. Statist. 16 (2) (1988) 772–783.

[6] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, Alexander Smola, A kernel two-sample test, J. Mach. Learn. Res. 13 (1) (2012) 723–773.

[7] Marco AF Pimentel, David A Clifton, Lei Clifton, Lionel Tarassenko, A review of novelty detection, Signal processing 99 (2014) 215–249.

[8] Christophe Leys, Olivier Klein, Yves Dominicy, Christophe Ley, Detecting multivariate outliers: Use a robust variant of the mahalanobis distance, J. Exp. Soc. Psychol. 74 (2018) 150–156.

[9] Roy De Maesschalck, Delphine Jouan-Rimbaud, Désiré L Massart, The mahalanobis distance, Chemometr. Intell. Laboratory Syst. 50 (1) (2000) 1–18.

[10] Eleazar Eskin, Anomaly detection over noisy data using learned probability distributions, 2000.

[11] Richard A. Redner, Homer F. Walker, Mixture densities, maximum likelihood and the EM algorithm, SIAM review 26 (2) (1984) 195–239.

[12] Melissa Turcotte, Juston Moore, Nick Heard, Aaron McPhall, Poisson factorization for peer-based anomaly detection, in: 2016 IEEE Conference on Intelligence and Security Informatics, ISI, IEEE, 2016, pp. 208–210.

[13] Weiming Hu, Jun Gao, Bing Li, Ou Wu, Junping Du, Stephen Maybank, Anomaly detection using local kernel density estimation and context-based regression, IEEE Trans. Knowl. Data Eng. 32 (2) (2018) 218–233.

[14] Gerhard Widmer, Miroslav Kubat, Learning in the presence of concept drift and hidden contexts, Mach. Learn. 23 (1) (1996) 69–101.

[15] Alexey Tsymbal, The problem of concept drift: definitions and related work, Computer Science Department, Trinity College Dublin 106 (2) (2004).

[16] João Gama, Pedro Medas, Gladys Castillo, Pedro Rodrigues, Learning with drift detection, in: Proceedings of 17th Brazilian Symposium on Artificial Intelligence - SBIA 2004, in: Lecture Notes in Computer Science, Springer, 2004, pp. 286–295.

[17] Manuel Baena-García, José del Campo-Ávila, Raúl Fidalgo, Albert Bifet, Ricard Gavaldà, Rafael Morales-Bueno, Early drift detection method, in: Proceedings of the Fourth International Workshop on Knowledge Discovery from Data Streams, number 6, 2006, pp. 77–86.

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

[19] Isvani Frias-Blanco, Jose del Campo-Avila, Gonzalo Ramos-Jimenez, Rafael Morales-Bueno, Agustin Ortiz-Diaz, Yaile Caballero-Mota, Online and non-parametric drift detection methods based on Hoeffding's bounds, IEEE Trans. Knowl. Data Eng. 27 (3) (2015) 810–823.

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

[21] Denis Moreira dos Reis, Peter Flach, Stan Matwin, Gustavo Batista, Fast unsupervised online drift detection using incremental Kolmogorov-Smirnov test, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16, 2016, pp. 1545–1554.

[22] Paul R. Rosenbaum, An exact distribution-free test comparing two multivariate distributions based on adjacency, J. R. Stat. Soc. Ser. B Stat. Methodol. 67 (4) (2005) 515–530.

[23] Daniel Kifer, Shai Ben-David, Johannes Gehrke, Detecting change in data streams, in: Proceedings of the 30th International Conference on Very Large Databases, vol. 30, Elsevier, 2004, pp. 180–191.

[24] Tamraparni Dasu, Shankar Krishnan, Suresh Venkatasubramanian, Ke Yi, An information-theoretic approach to detecting changes in multi-dimensional data streams, in: Proceedings of the Symposium on the Interface of Statistics, Computing Science, and Applications, 2006.

[25] Hayet Mouss, M.Djamel Mouss, Kinza Mouss, Sefouhi Linda, Test of Page-Hinckley, an approach for fault detection in an agro-alimentary production system, vol. 2, 2004, pp. 815–818.

[26] Ioannis Katakis, Grigorios Tsoumakas, Evangelos Banos, Nick Bassiliades, Ioannis Vlahavas, An adaptive personalized news dissemination system, J. Intell. Inf. Syst. 32 (2) (2008) 191–212.

[27] Ning Lu, Guangquan Zhang, Jie Lu, Concept drift detection via competence models, Artificial Intelligence 209 (2014) 11–28.

[28] Ning Lu, Jie Lu, Guangquan Zhang, Ramon Lopez de Mantaras, A concept drift-tolerant case-base editing technique, Artificial Intelligence 230 (2016) 108–133.

[29] Ben Shneiderman, Bridging the gap between ethics and practice: guidelines for reliable, safe, and trustworthy human-centered AI systems, ACM Trans. Interact. Intell. Syst. 10 (4) (2020) 1–31.

[30] Brian Dolhansky, Russ Howes, Ben Pflaum, Nicole Baram, Cristian Canton Ferrer, The deepfake detection challenge (dfdc) preview dataset, 2019, arXiv preprint arXiv:1910.08854.

[31] Markus Goldstein, Andreas Dengel, Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm, KI-2012: poster and demo track 1 (2012) 59–63.

[32] M.J. Desforges, P.J. Jacob, J.E. Cooper, Applications of probability density estimation to the detection of abnormal conditions in engineering, Proce. Inst. Mech. Eng. C: J. Mech. Eng. Sci. 212 (8) (1998) 687–703.

[33] Priyanga Dilini Talagala, Rob J. Hyndman, Kate Smith-Miles, Anomaly detection in high-dimensional data, J. Comput. Graph. Statist. 30 (2) (2021) 360–374.

[34] Shandong Wu, Brian E. Moore, Mubarak Shah, Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 2054–2060.

[35] Dihong Jiang, Sun Sun, Yaoliang Yu, Revisiting flow generative models for out-of-distribution detection, in: International Conference on Learning Representations, 2021.

[36] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, Haifeng Chen, Deep autoencoding gaussian mixture model for unsupervised anomaly detection, in: International Conference on Learning Representations, 2018.

[37] Erik Marchi, Fabio Vesperini, Florian Eyben, Stefano Squartini, Björn Schuller, A novel approach for automatic acoustic novelty detection using a denoising autoencoder with bidirectional LSTM neural networks, in: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2015, pp. 1996–2000.

[38] Pierre Baldi, Autoencoders, unsupervised learning, and deep architectures, in: Proceedings of ICML Workshop on Unsupervised and Transfer Learning, JMLR Workshop and Conference Proceedings, 2012, pp. 37–49.

[39] Diederik P. Kingma, Max Welling, Auto-encoding variational bayes, 2013, arXiv preprint arXiv:1312.6114.

[40] Davide Abati, Angelo Porrello, Simone Calderara, Rita Cucchiara, Latent space autoregression for novelty detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 481–490.

[41] Manassés Ribeiro, André Eugênio Lazzaretti, Heitor Silvério Lopes, A study of deep convolutional auto-encoders for anomaly detection in videos, Pattern Recognit. Lett. 105 (2018) 13–22.

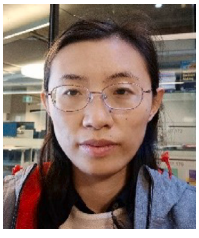
[42] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, Anil A Bharath, Generative adversarial networks: An overview, IEEE Signal Process. Mag. 35 (1) (2018) 53–65.

[43] Lucas Deecke, Robert Vandermeulen, Lukas Ruff, Stephan Mandt, Marius Kloft, Image anomaly detection with generative adversarial networks, in: Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2018, Dublin, Ireland, September 10–14, 2018, Proceedings, Part I 18, Springer, 2019, pp. 3–17.

[44] Anjin Liu, Yiliao Song, Guangquan Zhang, Jie Lu, Regional concept drift detection and density synchronized drift adaptation, in: IJCAI International Joint Conference on Artificial Intelligence, 2017.

[45] Tamraparni Dasu Shankar Krishnan Suresh Venkatasubramanian, Ke Yi, An Information-Theoretic Approach to Detecting Changes in Multi-Dimensional Data Streams.

[46] Alex Krizhevsky, Geoffrey Hinton, et al., Learning multiple layers of features from tiny images, 2009.



Dan Shang is a Ph.D student in Australian Artificial Intelligence Institute, University of Technology Sydney, Australia. Her research focuses on streaming data learning.

Guangquan Zhang is an Australian Research Council (ARC) QEII Fellow, Associate Professor and the Director of the Decision Systems and e-Service Intelligent (DeSI) Research Laboratory at the Australian Artificial Intelligence Institute, University of Technology Sydney, Australia. He received his Ph.D. in applied mathematics from Curtin University, Australia, in 2001. From 1993 to 1997, he was a full Professor in the Department of Mathematics, Hebei University, China. His main research interests lie in fuzzy multi-objective, bilevel and group decision making, fuzzy measures, transfer learning and concept drift adaptation. He has published six authored monographs and over 500 papers including some 300 articles in leading international journals. He has supervised 40 Ph.D. students to completion and mentored 15 Postdoc fellows. Prof Zhang has won ten very competitive ARC Discovery grants and many other research projects. His research has been widely applied in industries.

Jie Lu (F'18) is an Australian Laureate IEEE Fellow, IFSA Fellow, ACS Fellow, Distinguished Professor, and the Director of Australian Artificial Intelligence Institute (AAII) at the University of Technology Sydney, Australia. She received a Ph.D. degree from Curtin University in 2000. Her main research expertise is in transfer learning, concept drift, fuzzy systems, decision support systems and recommender systems. She has published over 500 papers in IEEE Transactions and other leading journals and conferences. She is the recipient of two IEEE Transactions on Fuzzy Systems Outstanding Paper Awards (2019 and 2022), NeurIPS2022 Outstanding Paper Award, Australia's Most Innovative Engineer Award (2019), Australasian Artificial Intelligence Distinguished Research Contribution Award (2022), Australian NSW Premier's Prize on Excellence in Engineering or Information & Communication Technology (2023), and the Officer of the Order of Australia (AO) 2023.