1
Introduction

Governments and companies are producing vast streams of data and require effective data analytics and machine learning methods to assist in making predictions and decisions promptly. One crucial aspect is the machine learning pipeline, which involves training a prepared dataset to construct a model and subsequently utilizing this model to predict new instance outputs. As depicted in Fig. 1.1, the process entails fetching historical data from the database during the training phase to construct the machine learning model. Then, the system can input new instances from the database to predict the output.

Nevertheless, when endeavoring to forecast outcomes for fresh instances sourced from an alternative database, as illustrated in Fig. 1.2 , there frequently emerges a conspicuous decline in accuracy. This disparity accentuates the imperative for model developers to intervene and rectify the issue. Addressing this, developers must adjust and retrain the model utilizing datasets from the new environment to ameliorate accuracy. This iterative process aims to refine the model's precision and ensure its efficacy across diverse contexts, thereby bolstering the reliability of

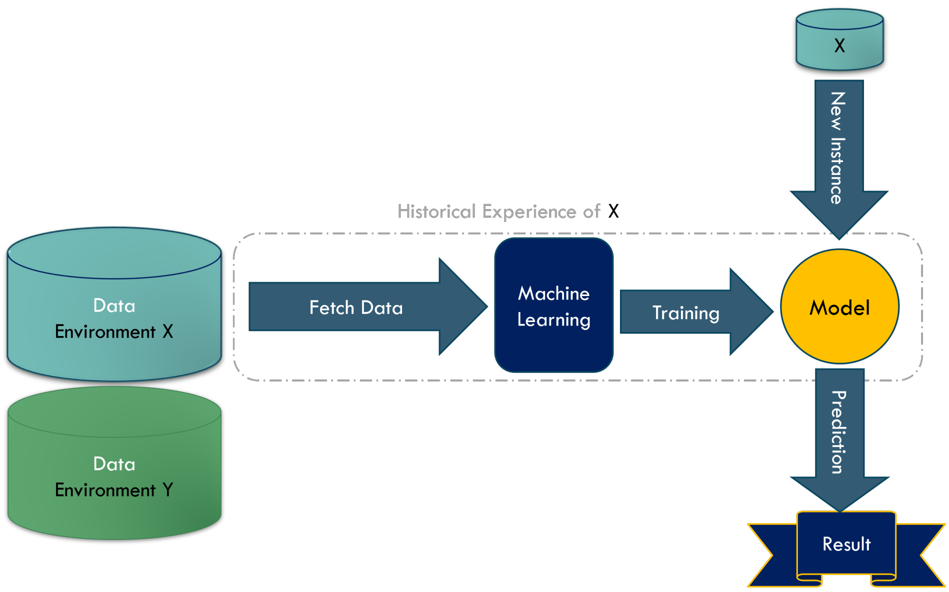


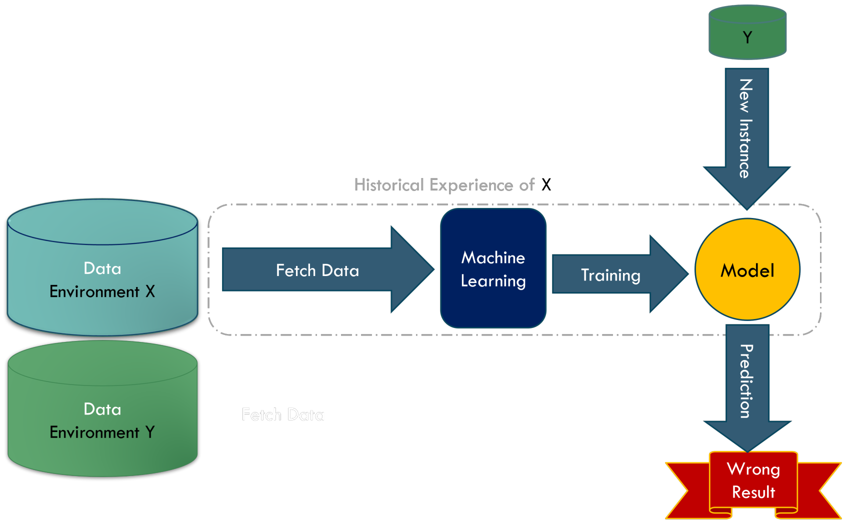
Figure 1.1: Machine learning workflow for environment X.

decision-making and predictive capabilities. To confront this challenge, the field of auto machine learning endeavors to facilitate online updates to the model without necessitating direct intervention from developers for modification.

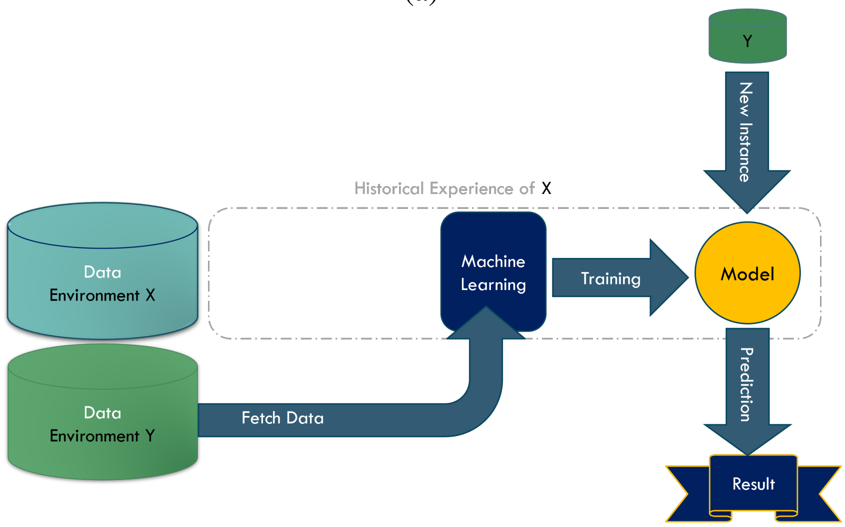
In recent years, high-speed data streams have presented significant challenges to machine learning models, particularly in streaming data analysis. These streams, characterized by continuous, dynamic, and high-volume data arrivals, demand adaptive learning algorithms to cope effectively with their evolving nature [1–3]. Among the critical challenges are concept drift, class imbalance, emerging new classes, and heterogeneous transfer learning.

Concept drift refers to changes in the statistical properties of data over time [4, 5], which necessitates continuous adaptation of machine learning models. This phenomenon may manifest as shifts in underlying concepts, relationships between variables, or alterations in data distribution. Addressing concept drift involves employing detection mechanisms that monitor classifier performance or data distribution changes, triggering model updates, retraining, or replacement. These dynamic adjustments ensure model efficacy in the face of evolving data streams.

Class imbalance and **class overlap** pose additional challenges, particularly in multi-class scenarios. Class imbalance, where data is unevenly dis-



(a)



(b)

Figure 1.2: Machine learning workflow for environment Y.

tributed across classes [6, 7], often leads to misclassification of underrepresented classes. Techniques such as oversampling, undersampling, algorithm adaptation, and hybrid approaches have been employed to address these challenges [8–11]. Class overlap, where instances from different classes occupy the same data regions [12, 13], complicates distinguishing between classes. Methods like class-overlap undersampling leverage local similarities to mitigate these issues.

Dynamic classifier ensembles, particularly **Dynamic Ensemble Selection (DES)** methods, offer solutions by adapting ensemble composition based on data characteristics [14]. These methods optimize classifier subsets through criteria like diversity metrics and performance estimation, ensuring a balance between accuracy and computational efficiency [15]. DES approaches can dynamically select the most competent classifiers for each data instance [16–18], enhancing classification in non-stationary streams.

Transfer learning plays a pivotal role in addressing dynamic data streams and concept drift, focusing on leveraging knowledge from source domains to improve target domain learning [4, 6]. Strategies include instance re-weighting, feature matching, and mitigating negative transfer effects to bridge domain gaps and improve adaptability.

Finally, the challenge of **Streams with Emerging New Classes (SENC)** involves scenarios where new classes, absent during initial training, emerge in data streams. Traditional models struggle to recognize and integrate these novel classes in real-time, necessitating adaptive learning mechanisms. This underscores the complexity of real-world data streams and highlights the importance of robust frameworks to manage such dynamics.

Overall, addressing these challenges requires integrating dynamic ensembles, adaptive sampling techniques, and transfer learning to ensure accuracy and scalability in non-stationary data environments. Figures and citations (e.g., Fig. 1 illustrating concept drift effects) support these discussions, emphasizing the critical need for adaptive solutions [19, 20].

In this chapter, the problem definition for this research that naturally arise is discussed in Section 1.1. After this, the the motivation is presented in Section 1.2 . After this, the objectives and research contribution are presented in Sections 1.3 and **??**, respectively. Next, the research contribution is summarised in Section 1.4. Finally, the outline of this thesis is presented in 1.6, respectively.

In this chapter, the problem definition for this research that naturally arise is discussed in Section 1.1. After this, the the motivation is presented in Section 1.2 . After this, the objectives and research contribution are presented in Sections 1.3 and 1.4. Next, the research plan is summarised in Section 1.5. Finally, the outline of this thesis is presented in 1.6, respectively.

1.1 Problem Definition

The rapid growth of high-speed data streams presents significant challenges for traditional machine learning models, particularly in non-stationary environments where data properties, concepts, and distributions evolve over time. These dynamic conditions cause models, initially trained on historical data, to experience a decline in accuracy when confronted with new data. The key challenges in managing non-stationary data streams are:

- **Concept Drift:** Changes in data distributions or relationships between variables over time, causing models to lose relevance and accuracy [1, 2].
- **Multi-class Imbalanced Data:** The uneven distribution of classes in multi-class data streams leads to biased predictions, with minority classes being misclassified [6, 7].
- **Class Overlap:** Instances from different classes occupying the same feature space, making it difficult for models to distinguish between them [12, 13].
- **Emergence of New Classes:** The appearance of new classes that were not present during training, causing instability and degraded performance as traditional models struggle to adapt.
- **Heterogeneous Transfer Learning:** The challenge of transferring knowledge between domains with differing characteristics, risking negative knowledge transfer in non-stationary data environments [4, 6].

These challenges hinder the performance and adaptability of machine learning models in real-time data streams, necessitating the development of advanced frameworks to address these issues effectively.

1.2 Thesis Motivations

The swift progress in machine learning, particularly in real-time data streams, introduces new challenges that demand innovative solutions. This thesis seeks to address the limitations of traditional models in dynamic, non-stationary data environments. The motivations behind this research are as follows:

- **Adapting to Emerging Classes in Real-Time:** New classes within data streams can cause a decline in accuracy. The goal is to develop adaptive mechanisms that quickly incorporate new classes, maintaining system relevance.

- **Proactive Management of Concept Drift:** Concept drift degrades model performance as data properties change. This research focuses on strategies to detect and manage concept drift to maintain model accuracy.

- **Dynamic Optimization of Classifier Ensembles:** To handle emerging classes and shifting distributions, we aim to develop techniques for real-time optimization of classifier ensembles.

- **Addressing Multi-Class Imbalance:** Imbalanced multi-class distributions often lead to biased classification. The thesis aims to create methods that address class imbalances dynamically, ensuring fair classification.

- **Enhancing Transfer Learning in Non-Stationary Environments:** Transfer learning can suffer from negative transfer in non-stationary settings. The research focuses on frameworks that minimize negative transfer and enhance knowledge sharing across diverse domains.

1.3 Thesis Objectives

The thesis objectives for our research stems from the imperative need to address the following key motivations:

- **Objective 1**. Handling Imbalanced Multiclass Drifted Data and overlapping classes streams.
- **Objective 2**. Addressing Emerging New Classes in Incremental Streams via Concept Drift and K-means Techniques.
- **Objective 3**. Addressing Heterogeneous Transfer Learning Problem in data streams via Concept Drift and Eigenvector Techniques.

1.4 Contributions

Our research focuses on developing advanced frameworks to manage non-stationary data streams with high accuracy and minimal computational complexity. The key contributions include:

- **Concept Drift Detection and Ensemble Classifier:** Integrating concept drift detection with ensemble classifiers for real-time adaptation in transfer learning.
- **Dynamic Classifier Ensemble for Emerging Classes:** A framework that dynamically adjusts classifiers to address emerging class issues in non-stationary streams.
- **Precise Weighting for Local Classifiers:** A novel method to refine local classifier contributions, enhancing overall ensemble accuracy.
- **Eigenvector-Based Framework for Heterogeneous Transfer Learning:** A framework using eigenvectors to facilitate knowledge transfer across diverse domains, improving performance.
- **Dynamic Adjustment for Multi-class Imbalanced Data:** A method combining drift detection and optimized ensemble selection to improve accuracy in imbalanced streams.
- **Adaptive Class Imbalance Method:** A dynamic approach to select oversampling methods, addressing class overlap in drifted data streams.

1.5 Research Plan

The research in this thesis is progressing rapidly due to technological advancements and continuous contributions in machine learning (ML). An iterative research methodology was followed, where each cycle builds upon the knowledge gained in the previous phase, leading to increasingly effective and original solutions as shown in Fig. 1.3. The phases of this research methodology are as follows:

1. **Review of the current state-of-the-art:** Investigate existing research to identify challenges and inform the design of a solution.

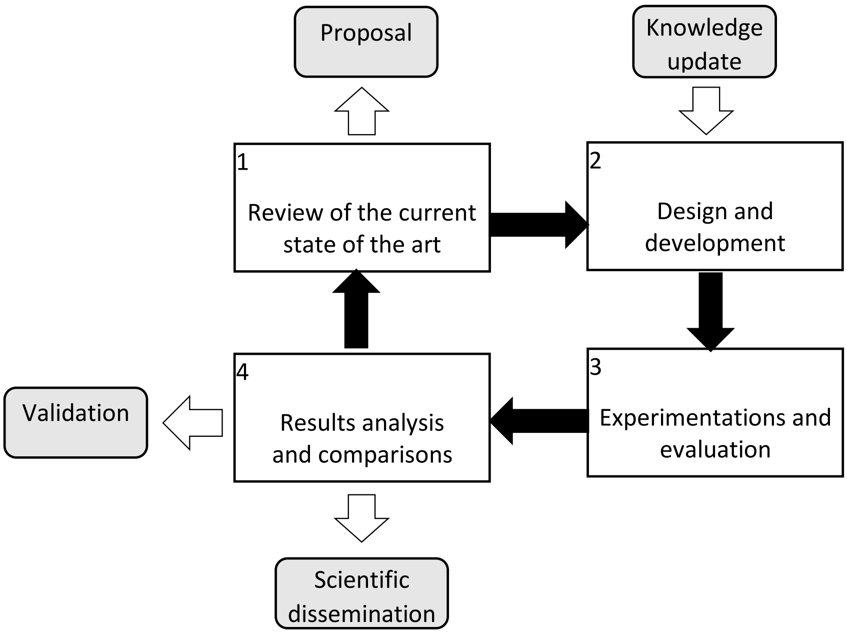


Figure 1.3: The research methodology of the thesis.

2. **Design and development:** Design a novel solution using updated knowledge to address the identified challenges.

3. **Experimentation and evaluation:** Test the solution through experimentation, using established criteria for comparison.

4. **Results analysis and comparison:** Analyze and compare results with state-of-the-art to determine the effectiveness of the solution, and disseminate the findings.

1.6 Structure of the Dissertation

The structure of the remainder of this thesis dissertation is outlined below:

- **Chapter 2** reviews background about concept drift, concept drift types, concept drift components, adapting types.
- **Chapter 3** reviews state-of-the-art concept drift, classifier ensemble selection, imbalanced data streams, Streams with Emerging New Classes (SENC), and transfer learning.

- **Chapter ??** presents our first proposal to build an effective proposed approch for handling Imbalanced Multi-class Drifted Data streams.
- **Chapter ??** provides our second proposal to adressing emerging new classes in incremental streams via concept drift techniques.
- **Chapter ??** presents our third proposal to addressing heterogeneous transfer learning problem in incremental streams via concept drift techniques.
- **Chapter ??** revisits the main goal and specific objectives posted earlier. In this chapter, we summarise the main contributions of this thesis and outline possible future research.

2 Background

Amidst the surge of vast streaming data, governments and businesses find themselves in an urgent need for sophisticated data analysis and machine learning analytics approaches. These tools are indispensable for anticipating future trends and making well-informed decisions. However, the perpetual emergence of new goods, markets, and consumer behaviors introduces a formidable challenge known as concept drift [21]. This phenomenon involves the variation of statistical parameters of the target variable over time in unexpected ways, posing a substantial obstacle to accurate forecasting and optimal decision-making. The patterns derived from historical data may become obsolete when applied to new and evolving datasets. The impact of concept drift extends across data-driven information systems, including decision support and early warning systems, diminishing their overall effectiveness. In the dynamic realm of big data, where data types and distributions are inherently unpredictable, the challenge of concept drift becomes even more pronounced. In response to this challenge, the field introduces a new subject: adaptive data-driven prediction/decision systems.

2.1 Concept Drift Sources

Concept drift, illustrated comprehensively in Fig. 2.1, unfolds through three distinct scenarios. First, (a) portrays a shift in data distribution, signifying changes in the underlying patterns and characteristics of the incoming data. This type of drift challenges the model's ability to adapt to new trends and patterns [22–25]. Second, (b) showcases a change in function output, leading to the need for adjustments in the position of the class delimiter. Here, the relationship between input features and output classes undergoes transformations, demanding the model to realign its decision boundaries accordingly. Third, (c) presents a scenario involving a dual shift—both in data distribution and function output. This complex manifestation of concept drift requires the model to address simultaneous changes in underlying patterns and output relationships. Effectively navigating these shifts is crucial for maintaining the model's predictive accuracy and decision-making capabilities in dynamic environments. Understanding these nuanced aspects of concept drift is foundational for devising adaptive strategies in machine learning models.

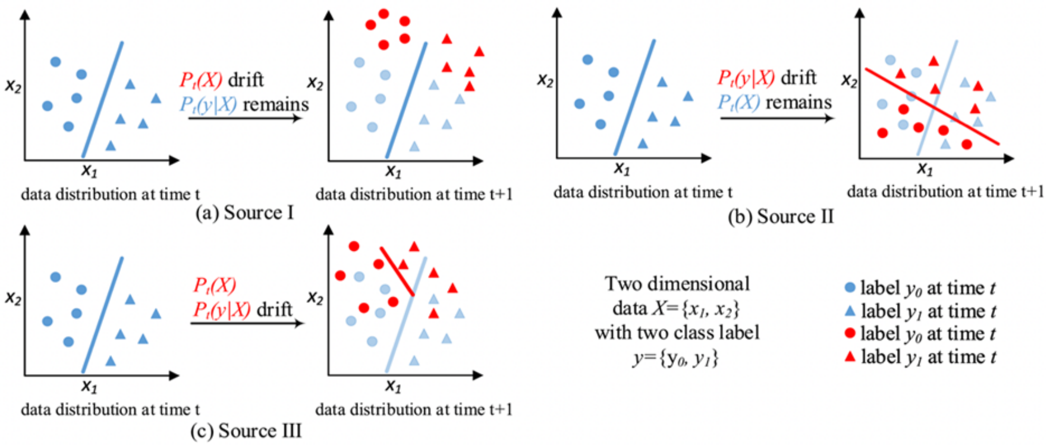


Figure 2.1: Sources of concept drift.

DOI: 10.1109/TKDE.2018.2876857

2.2 Concept Drift Types

Concept drift is commonly categorized into four types, as illustrated in Fig. 2.2. In Types 1-3, the focus of research on concept drift adaptation revolves around minimizing the drop in accuracy and achieving the fastest recovery rate during the transformation process of concepts. Conversely, Type 4 drift delves into leveraging historical concepts, emphasizing the identification of the best-matched historical concepts within the shortest time frame when a new concept emerges—whether suddenly, incrementally, or gradually. To illuminate the distinctions between these types, [24] introduces the term ”intermediate concept” to describe the transitional phases between concepts. As noted by [11], concept drift may not only occur at a precise timestamp but may also extend over a prolonged period. Consequently, intermediate concepts may emerge during the transformation, representing a blend of the starting and ending concepts in the case of incremental drift or embodying one of the starting or ending concepts, as observed in gradual drift. Understanding these intermediate concepts is essential for comprehending the nuanced dynamics of concept drift during transitions from one state to another.

2.3 Concept Drift Components

Conventional machine learning primarily involves prediction and training/learning. However, learning under the concept drift paradigm introduces three additional critical steps as show in Fig. 2.3: concept drift detection, drift understanding, and drift adaptation. Concept drift detection involves identifying changed points or change time periods to define and predict drift. Drift understanding delves into crucial aspects such as when the drift starts, how long it lasts, and where it occurs, providing indispensable insights for the subsequent adaptation step. The adaptation step, also referred to as the reaction step, plays a pivotal role in updating current learning models in response to concept drift. Three main approaches address various types of drift: Simple Retraining, Ensemble Retraining, and Model Adjusting. Drift detection employs various tools and algorithms, comparing old and fresh data chunks with statistical models based on data distribution. Techniques vary, with some

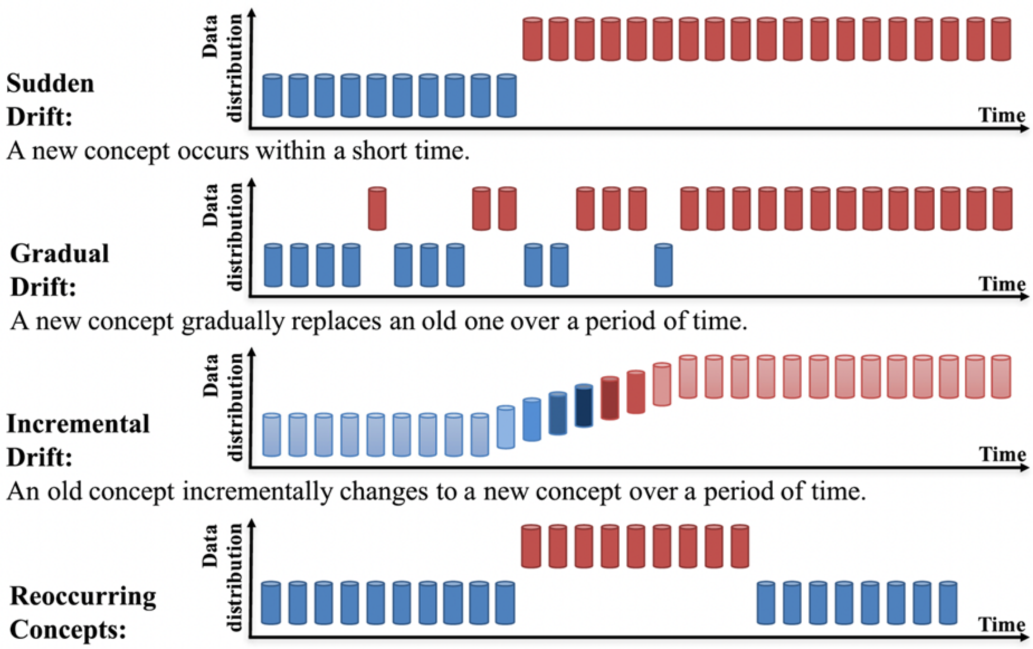


Figure 2.2: Types of concept drift.

DOI: 10.1109/TKDE.2018.2876857

utilizing a constant chunk length and others employing a variable length. Drift understanding is essential for making well-informed decisions during the adaptation step. This involves calculating the necessary modifications in the trained model to adapt to new changes as shown in Fig. 2.4. The severity region determines whether to generate a completely new model or make minimal adjustments to the existing one.

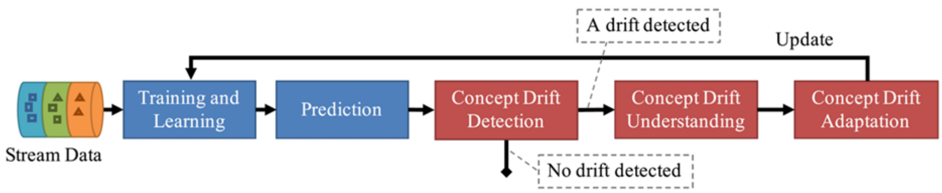


Figure 2.3: Main components of concept drift.

DOI: 10.1109/TKDE.2018.2876857

2.3.1 Concept Drift Detection

Drift detection involves techniques and mechanisms to characterize and quantify concept drift by identifying change points or intervals [11]. The general framework for drift detection consists of four stages:

- **Stage 1 (Data Retrieval):** This stage focuses on retrieving data chunks from data streams. Given that a single data instance lacks sufficient information to infer the overall distribution [22], organizing data chunks meaningfully is crucial for effective data stream analysis [26].

- **Stage 2 (Data Modeling):** This optional stage abstracts the retrieved data, extracting key features that contain sensitive information impacting a system in case of drift. This stage may involve dimensionality reduction or sample size reduction to meet storage and online speed requirements [11].

- **Stage 3 (Test Statistics Calculation):** This stage involves measuring dissimilarity or estimating distance to quantify drift severity and generate test statistics for hypothesis testing. Defining an accurate and robust dissimilarity measurement remains a challenging aspect of concept drift detection. Test statistics can also be used for clustering evaluation [27] and to determine dissimilarity between sample sets [28].

- **Stage 4 (Hypothesis Test):** This stage employs a specific hypothesis test to assess the statistical significance of the change observed in Stage 3, such as the p-value. These tests determine drift detection accuracy by establishing statistical bounds for the test statistics from Stage 3. Without Stage 4, the acquired test statistics are meaningless for drift detection, as they cannot establish the drift confidence interval. Commonly used hypothesis tests include estimating the distribution of test statistics [29, 30], bootstrapping [31, 32], the permutation test [22], and Hoeffding's inequality-based bound identification [33].

It is crucial to note that without Stage 1, the concept drift detection problem can be regarded as a two-sample test problem, examining whether the populations of two given sample sets are from the same distribution [28]. In other words, any multivariate two-sample test can be adopted in Stages 2-4 for detecting concept drift [28]. However, in cases where distribution drift may not be included in the target features, the selection of the target feature becomes critical for the overall performance of a learning system and poses a significant challenge in concept drift detection [34].

2.3.2 Understanding Phase

The extent of concept drift severity serves as a valuable criterion for selecting appropriate drift adaptation strategies. As shown in Fig. 2.4, in a classification task where the drift's severity is minimal, resulting in only a marginal shift in the decision boundary within the new concept, adjusting the current learner through incremental learning proves sufficient. Conversely, when confronted with high severity in concept drift, wherein the decision boundary undergoes substantial changes, it might be more effective to discard the old learner and opt for retraining a new one rather than incrementally updating the existing learner. It's noteworthy to mention that despite some researchers highlighting the capability to articulate and quantify the severity of detected drift, this information is not yet widely integrated into drift adaptation practices. The adaptation step offers three distinct ways to adapt the model. Simple Retraining involves training a new model using the most recent data, replacing the old model. Model Ensemble, the second approach, entails keeping and reusing existing models, which proves efficient when dealing with repeated instances of concept drift. The third approach, Model Adjusting, constructs a model that adapts flexibly from changed data, allowing partial updates when the original data distribution undergoes significant changes.

2.3.3 Adaptation Phase

This section delves into strategies for updating existing learning models in response to drift, referred to as drift adaptation or reaction. The three primary categories of drift adaptation methods are simple retraining, ensemble retraining, and model adjusting, each tailored to address specific types of drift.

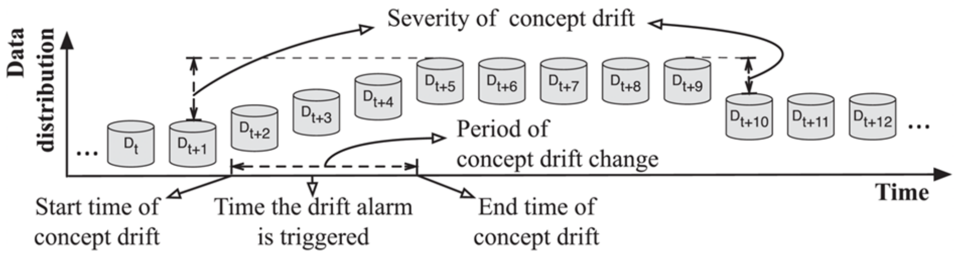


Figure 2.4: Understanding phase of concept drift.
DOI: 10.1109/TKDE.2018.2876857

A. **Simple Retraining**
One way to respond to concept drift is by retraining a new model with the latest data, replacing the outdated model as shown in Fig.2.5. This approach requires an explicit concept drift detector to determine when to retrain the current model. A window strategy is commonly used, preserving recent data for retraining and/or utilizing old data for distribution change tests. An example of this strategy is Paired Learners [35], which employs two learners: the stable learner and the reactive learner. If the stable learner consistently misclassifies instances correctly classified by the reactive learner, indicating a new concept, the stable learner is replaced with the reactive learner. This method is straightforward, easy to implement, and adaptable at any point in the data stream. However, a trade-off arises when adopting a window-based strategy in determining an appropriate window size. A small window better reflects the latest data distribution, while a large window provides more data for training a new model. To address this challenge, the ADWIN algorithm [36] dynamically adjusts sub window sizes based on the rate of change between two sub-windows, eliminating the need for users to predefine window sizes. Beyond direct model retraining, researchers have explored integrating the drift detection process with the retraining process for specific machine learning algorithms. DELM [37], for instance, extends the traditional ELM algorithm, adapting to concept drift by dynamically adjusting the number of hidden layer nodes in response to increasing classification error rates, indicating a potential concept drift. Similarly, FP-ELM [38] introduces a forgetting parameter to the ELM model to adapt to drift conditions. A parallel version

of the ELM-based method [39] has been developed for high-speed classification tasks under concept drift. OS-ELM [40], an online learning ensemble of repressor models, integrates ELM using an ordered aggregation (OA) technique to address the challenge of defining the optimal ensemble size. In the realm of instance-based lazy learners for handling concept drift, the Just-in-Time adaptive classifier [29] follows the ”detect and update model” strategy, extending the traditional CUSUM test [41] to a pdf-free form for drift detection. When a concept drift is identified, old instances (beyond the last T samples) are removed from the case base. Advancements include extending this algorithm to handle recurrent concepts by considering and comparing the current concept to previously stored concepts [27, 29]. NEFCS [22], another KNN-based adaptive model, employs a competence model-based drift detection algorithm [22] to locate drift instances in the case base and distinguish them from noise instances. The redundancy removal algorithm, Stepwise Redundancy Removal (SRR), is developed to eliminate redundant instances uniformly, ensuring that the reduced case base retains sufficient information for future drift detection.

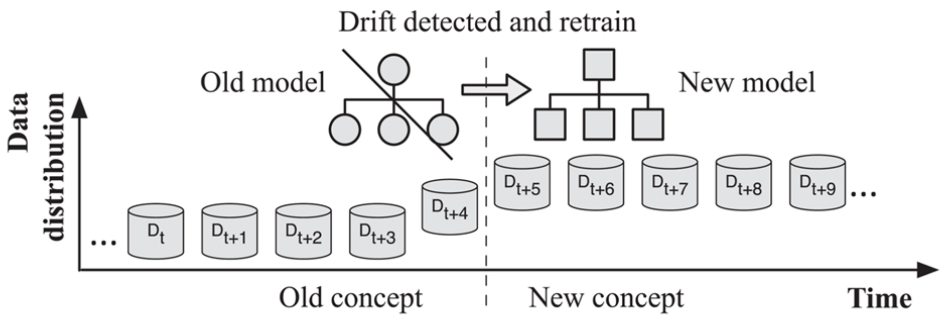


Figure 2.5: Approach for retraining a new model.
DOI: 10.1109/TKDE.2018.2876857

B. **Model Ensemble Eetraining**

In recurring concept drift scenarios, preserving and reusing old models can be more efficient than retraining new ones for each recurrence, forming the basis for employing ensemble methods [42]. These methods, a focus in stream data mining research, consist of base classifiers with varied types or parameters. Their outputs, determined by specific voting rules, collectively predict new data. Various adaptive ensemble methods, extending classical ones or introducing adaptive voting rules, address the challenges of concept drift as shown in Fig. 2.6. Classical ensemble methods like Bagging, Boosting, and Random Forests have been adapted for streaming data with concept drift. For instance, online Bagging [43] uses each instance once, simulating batch mode bagging. Leveraging Bagging [44] employs ADWIN drift detection to replace the existing classifier with the worst performance when concept drift is detected. Adaptive boosting [45], monitoring prediction accuracy through a hypothesis test, addresses concept drift, assuming classification errors on non-drifting data follow a Gaussian distribution. The Adaptive Random Forest (ARF) algorithm [46] extends the random forest tree algorithm, incorporating concept drift detection (e.g., ADWIN) to decide when to replace an obsolete tree. A similar approach is seen in [47], using Beyond classical methods, novel ensemble methods with innovative voting techniques tackle concept drift. Dynamic Weighted Majority (DWM) [48] adapts to drifts through weighted voting rules, managing base classifiers based on individual and global ensemble performance. Learn++NSE [49] addresses frequent classifier additions by weighting them based on prediction error rates. Specific types of concept drift are considered in specialized ensemble methods. Accuracy Update Ensemble (AUE2) [50] equally addresses sudden and gradual drift, using a batch mode weighted voting ensemble method. Optimal Weights Adjustment (OWA) [51] achieves a similar goal with weighted instances and classifiers. Special cases, like class evolution, are considered in [52], while recurring concepts are handled by monitoring concept information [23, 53]. Another method [54], refines the concept pool for recurring concepts.

C. **Model Adjusting**

As Shown in Fig. 2.7, instead of retraining the entire model, an alternative is to construct a model with adaptive learning capabilities, allowing partial updates in response to changing data distributions [55], as in Fig. 2.7. This is efficient when concept drift occurs in localized regions. Many techniques

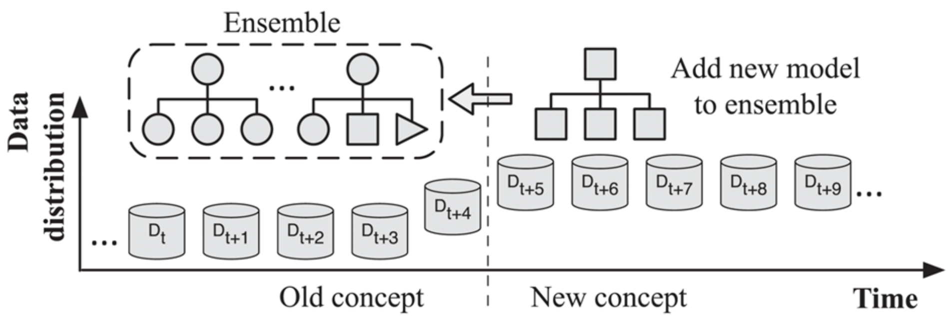


Figure 2.6: Ensemble approach for the adaptation phase.
DOI: 10.1109/TKDE.2018.2876857

in this category use the decision tree algorithm, leveraging its ability to adapt to individual sub-regions. VFDT [56] is a foundational contribution for high-speed data streams, employing the Hoeffding bound for node splitting. VFDT processes each instance once, doesn't store instances, and has minimal maintenance costs. CVFDT [57], an extension, addresses concept drift by maintaining a sliding window of the latest data and replacing the original sub-tree with a better-performing alternative. VFDTc [57] enhances VFDT by handling numerical attributes and adapting to concept drift with node-level detection. Later extensions [58, 59] introduce an adaptive leaf strategy in VFDTc, selecting the best classifier from options like majority voting, Naive Bayes, and Weighted Naive Bayes. Recent studies [60, 61] question VFDT's foundation, the Hoeffding bound, for non-independent variables in information gain. An alternative impurity measure is proposed in a new online decision tree model [61], demonstrating its reflection of concept drift and potential use in CVFDT. IADEM-3 [62] addresses Hoeffding bound concerns by computing the sum of independent random variables for drift detection and pruning.

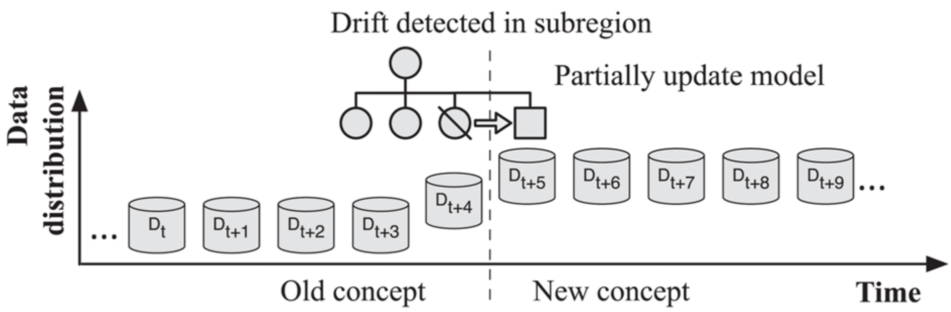


Figure 2.7: Partial updating approach for the adaptation phase.

DOI: 10.1109/TKDE.2018.2876857

3

State-of-the-art

In this chapter, we provide a comprehensive review of recent advancements in stream classification, addressing several critical challenges that arise in dynamic data environments. The rapid evolution of data streams introduces complexities such as concept drift, imbalanced multiclass scenarios, class overlap, classifier ensemble selection, emergence of new classes, and the incorporation of transfer learning. This introduction outlines the structure of the chapter and highlights the significance of each topic in advancing stream classification methodologies. The first section (Setion 3.1) focuses on concept drift, a phenomenon where the statistical properties of the data change over time. We explore various methodologies developed for real-time detection and adaptation to concept drift in streaming scenarios. This includes a review of techniques that enable systems to identify shifts in data patterns promptly, ensuring sustained classification accuracy. By analyzing the strengths of these approaches, we highlight the necessity for robust drift detection mechanisms in evolving data streams. Next, in Section 3.2, we delve into classifier ensemble selection in the context of streaming data. As the characteristics of data streams evolve, selecting the most appropriate ensemble of classifiers becomes crucial. We present algorithms designed to dynamically choose the best-performing classifiers based on the current data distribution. This section emphasizes the importance of adaptive ensemble strategies in enhancing classification performance amidst changing data landscapes. Section 3.3 addresses the challenges posed by imbalanced multiclass scenarios, particularly in the presence of overlapping classes. We discuss specialized oversampling techniques aimed at countering the effects of class imbalance in streaming data. By providing insights into effective strategies for balancing the representation of minority classes, we lay the groundwork for developing robust frameworks that can maintain performance in imbalanced drifted streams. As new classes emerge in streaming systems, traditional classifiers often struggle to adapt effectively. In Section 3.4, we explore methodologies that specifically focus on the integration of new classes within existing frameworks. This section highlights innovative approaches that enhance the adaptability of stream classification systems, ensuring they remain effective as new data patterns emerge. In Section 3.5, we examine the role of transfer learning in stream classification. This section discusses how transfer learning techniques can leverage knowledge from related tasks to improve classification performance in dynamic environments. We explore current methodologies that facilitate the transfer of information across different domains, particularly in situations where labeled data may be scarce. The insights gained from this discussion will be pivotal for understanding how transfer learning can complement other strategies in our proposed framework. In Section 3.6,To further contextualize our discussion, we conduct a comparative analysis of recent works in the field, focusing on the recent works for each challange. We assess their contributions and identify limitations, illuminating specific gaps in the current research landscape that our work aims to address. Finally, Section 3.7 concludes this chapter by identifying key challenges and gaps, which inform the research direction for the subsequent chapters.

3.1 Concept Drift

In machine learning, concept drift refers to the phenomenon where the underlying data distribution changes over time [63–65] leading to a decrease in the accuracy or relevance of previously trained models. Detecting and responding to concept drift promptly is crucial for maintaining the performance of machine learning models. To address this challenge, various concept drift detection methods have been proposed in the literature. One widely used method is the Drift Detection Method (DDM)[30, 44] which employs a statistical test to compare the error rate of a model on consecutive data sets. By identifying significant performance decreases, DDM signals the presence of concept drift. Another approach is the Early Drift Detection Method (EDDM)[30, 66] an extension of DDM that considers the error rate of a moving window of the latest data compared to the previous window. The ADaptive WINdow (ADWIN)[30, 66] approach uses a sliding window technique to monitor statistical differences between consecutive windows for drift detection. It dynamically adjusts the window size to adapt to changing drift patterns. The Kolmogorov-Smirnov windowing method (KSWIN) [66] calculates the Kolmogorov-Smirnov distance between two sliding windows to detect concept drift. Hoeffding's bounds with moving average test (HDDMA) and its variant HDDMW compute upper and lower bounds for the true mean of the data stream [30, 44] . By comparing these bounds, they can detect changes in the data distribution indicative of concept drift. Lastly, the Page-Hinkley [67] method measures the cumulative sum of errors and detects drift when the sum exceeds a predefined threshold. These concept drift detection methods are crucial in enabling machine learning models to adapt to the evolving nature of data streams. By continuously monitoring and detecting changes in data distributions, these methods facilitate the necessary adjustments and updates to maintain the model's performance and accuracy over time. Incorporating these methods into machine learning frameworks enhances their robustness and enables them to handle concept drift effectively.

3.2 Classifier Ensemble Selection

This study focuses on the overproduce-and-select approach for classifier ensemble selection methods [15, 18, 68]. The primary objective of classifier ensemble selection is to identify the optimal subset of classifiers from a larger ensemble, considering various criteria such as performance measures, diversity metrics, meta-learning techniques, and performance estimation approaches. This selection process aims to reduce computational complexity, enhance efficiency, and improve overall ensemble performance, making it highly valuable for real-world applications. By carefully selecting a smaller subset of classifiers, ensemble selection strikes a balance between accuracy and computational resources, adapting to the evolving nature of the data stream. This approach leverages the strengths of different classifiers and adjusts the ensemble composition to handle changing conditions effectively. The goal is to enhance the accuracy, robustness, and overall performance of classification models in dynamic and challenging scenarios. There are two main approaches to the selection process: static and dynamic selection. Static selection assigns classifiers to specific partitions of the feature space, while dynamic selection chooses a classifier specifically for each unknown data sample based on its local competencies. Dynamic Ensemble Selection (DES) is a widely recognized approach that selects the best classifiers for each test instance, considering their competence within the local region of competence. The Randomized Reference Classifier proposed by Woloszynski and Kurzynski [16] stands out among various approaches. This classifier introduces randomness through beta distribution, enhancing adaptability and robustness. By considering the stochastic nature of class supports, the Randomized Reference Classifier can potentially improve classification performance in concept drift scenarios. However, it is important to note that employing diversity measures during the classifier selection process, as demonstrated by Lysiak [17], may lead to smaller ensembles but does not necessarily enhance classification accuracy. Overall, the overproduce-and-select approach for classifier ensemble selection methods offers a comprehensive framework for addressing the challenges associated with concept drift. By dynamically adapting the ensemble composition and leveraging the competencies of individual classifiers, this approach aims to improve classification performance, efficiency, and adaptability in dynamic and challenging scenarios.

3.3 Imbalanced data Streams

In the context of imbalanced data classification, as previously discussed, researchers have categorized three primary methodologies[69]. Our study primarily focuses on the first category, which pertains to addressing imbalanced data streams, particularly through sampling methods, specifically generating synthetic instances to rectify the class imbalance. This process is commonly referred to as oversampling[70] and aims to balance instance quantities across both classes [71–74] . It's important to note that class imbalance can manifest in two scenarios: binary imbalanced classes, featuring skewed distribution between two classes, and multi-class imbalanced situations. Our study is specifically centered on multi-class oversampling techniques. In addressing class imbalances within multi-class contexts, Multi-Label SMOTE (MLSMOTE) [9] has extended the principles of SMOTE to cater to imbalanced multi-class learning scenarios. MLSMOTE significantly enhances classifier performance by generating synthetic examples for each minority class label. This approach thoughtfully considers neighboring examples in the feature space, ensuring that synthetic examples are appropriately assigned to their respective minority classes. Moreover, a recent advancement known as Multi-Label Synthetic Oversampling based on Local label imbalance (MLSOL) [69] has emerged as a promising technique. MLSOL systematically combats local imbalances within the domain of multi-class classification by employing distinct sampling strategies for each label. Empirical research confirms MLSOL's superiority by showcasing its capability to outperform existing methods in terms of classification accuracy and other evaluation metrics, particularly in effectively addressing local imbalance. Importantly, MLSOL offers a potentially effective approach to enhancing the performance of multi-class classification models, surpassing MLSMOTE in several aspects. Notably, MLSOL constructs synthetic samples exclusively from minority class instances within a restricted neighborhood, resulting in a more compact synthetic dataset compared to MLSMOTE. This attribute bears potential advantages for computational efficiency and helps mitigate concerns related to overfitting.

3.4 Streams with Emerging New Classes (SENC)

Existing approaches have been proposed to detect and handle the emergence of new classes in streaming data. Clustering-based methods, such as SACCOS [75], EC-SMiner [76], and SAND [77], employ clustering techniques to identify new class emergence. However, these methods require access to true labels for either parts or all instances, limiting their practical applicability. Similarly, SENC-MaS [78] uses matrix sketches for detecting emerging new classes but assumes the availability of true label information for all instances. In contrast, tree-based methods like SENCForest [79] and SEEN [80] utilize anomaly detection techniques to identify new classes, often with limited or no label information. However, these methods often suffer from high false positive rates and runtime inefficiencies. Another approach, SENNE [81], focuses on exploiting local information using the nearest neighbor ensemble for improved detection performance. Nevertheless, the absence of an effective model retirement mechanism in SENNE results in longer runtimes than alternative methods. The k-nearest Neighbor Ensemble-based method (KNNENS) [82] method emerges as a promising solution for the challenges of streaming emerging new class problems. By effectively utilizing a k-nearest neighbor-based hypersphere ensemble and incorporating model updates, the KNNENS approach tackles the issues of new class detection and known class classification within a unified framework. It is worth noting that an explicit limitation of existing methods is their lack of utilization of concept drift techniques for detecting emerging new classes and retraining the classification model. This limitation highlights the need for approaches that can effectively handle concept drift while addressing the emergence of new classes in streaming data.

3.5 Transfer Learning

In recent years, transfer learning has received significant attention due to its growing importance in addressing disparities between source and target domain distributions. Bridging the gap in distribution disparities is vital for optimizing performance in transfer learning, resulting in the development of diverse approaches, which can be broadly categorized into instance re-weighting and feature matching [83]. Instance re-weighting methods focus on aligning domain distributions by adjusting the weights of source instances, enabling the reuse of those source instances that closely align with the target domain. Notably, there's a strong emphasis on estimating these instance weights. For example, Huang et al. [84] introduced the Kernel Mean Matching (KMM) technique, which calculates weights by minimizing the mean differences between instances from the source and target domains within a Reproducing Kernel Hilbert Space (RKHS). Sugiyama et al. [85] put forth the Kullback-Leibler Importance Estimation Procedure (KLIEP), utilizing Kullback-Leibler distance as a metric for assessing domain distribution dissimilarity, which includes a model selection step. Building on these instance reweighting methods, Sun et al. [86] introduced the 2-Stage Weighting Framework for Multisource Domain Adaptation (2SW-MDA) to address challenges in multisource transfer learning. It simultaneously adjusts the weights of source domains and their instances to reduce both marginal and conditional distribution disparities, akin to KMM, while leveraging the smoothness assumption for domain weighting. TrAdaBoost [87], a variation of the AdaBoost framework [88], operates by iteratively adjusting the weights of training data. In each iteration, it trains a classifier on a mix of source and target data and uses this classifier to make predictions on the training data. If a source instance is incorrectly predicted, its weight is reduced, diminishing its influence on the classifier. Conversely, the weights of misclassified target instances are increased to amplify their impact. An extension of TrAdaBoost, known as Multisource TrAdaBoost (MsTrAdaBoost) [89], is employed to address multisource transfer learning challenges. MsTrAdaBoost combines each source and target dataset, training a separate classifier for each. Subsequently, it selects the classifier with the least error on the target data to update the instance weights. On a different note, feature matching aims to establish a shared feature representation space between source and target domains, which can be achieved through either symmetric or asymmetric transformations. A typical example of a symmetric transformation is the Transfer Component Analysis (TCA) method by Pan et al. [89], employing Maximum Mean Discrepancy (MMD) [90, 91] [83] to minimize differences in marginal distribution between source and target domains within an RKHS. Expanding on TCA, Joint Distribution Adaptation (JDA) [92] has been introduced to address both marginal and conditional distribution disparities. Recognizing the varying importance of marginal and conditional distribution differences across different problems, Wang et al. [92, 93] introduced the Balanced Distribution Adaptation (BDA) approach, which introduces a balancing factor. Subspace Alignment (SA) [94] focuses on aligning domain distributions in a lower-dimensional subspace, selecting crucial eigenvectors using principal component analysis [94] and learning a linear transformation matrix to minimize differences in eigenvectors between domains. In contrast, the Distribution Alignment between Two Subspaces (SDA-TS) [95] was proposed to align both bases and distributions. Correlation Alignment (CORAL) [96], [84], an asymmetric transformation approach, is designed to align sub-space bases and employs second-order statistics. CORAL uses a learned transformation matrix to project source instances into the target domain. While there is a wide range of feature matching techniques in transfer learning, it is imperative to prevent the negative transfer, which occurs when transferred knowledge hinders the performance of target tasks. One of the reasons for negative transfer is the inclusion of unrelated or detrimental source samples in the target domain. To mitigate this, transfer joint matching (TJM) [97] introduces sparsity regularization in the feature transformation matrix, aligning features and re-weighting instances simultaneously. Zhong et al. [98] have also developed strategies to mitigate the impact of unrelated source instances and ensure positive transfer. To tackle the challenges posed by partial transfer learning scenarios, where the source domain contains more classes than the target domain, prior works [1, 99, 100] introduce instance-level re-weighting and class-level re-weighting mechanisms. These mechanisms are employed to reduce the influence of outlier classes from the source domains. Additionally, another approach presented in [3] utilizes an adversarial neural network to align domain distributions. In this method, lower weights are assigned to the source samples that are deemed distant from the discriminator. This weighting reflects the perception that such instances have weaker relevance to the target domain. Yang et al. [101] introduce the HE-CDTL approach for Concept Drift Transfer Learning (CDTL). HE-CDTL leverages knowledge from both source domains and historical time steps within the target domain to improve learning performance. Its key advantages include the utilization of the class-wise weighted ensemble for historical knowledge and the implementation of AW-CORAL for knowledge extraction from source domains. The class-wise weighted ensemble empowers individual classes in the current learning process to select historical knowledge independently. AW-CORAL serves to minimize domain disparities between source and target domains while mitigating negative knowledge transfer. Extensive experiments demonstrate that HE-CDTL outperforms baseline methods in addressing transfer learning challenges in the context of concept drift. Melanie addresses the challenge of non-stationary environments by considering an online scenario where data in both source and target domains are generated. This method employs an online ensemble to learn models from each domain, subsequently combining these models using a weighted-sum approach. The models are trained incrementally, with their weights dynamically adjusted to handle concept drift. Generally, Melanie can be adapted to address CDTL by substituting the online learning ensemble with an ensemble designed for chunk-based concept drift.

3.6 Comparsion

In this section, we undertake a critical comparison of closely related works addressing the challenges of imbalanced multiclass streams 3.6.1, the emergence of new classes 3.6.2, and the integration of transfer learning 3.6.3 within streaming environments. The increasing complexity of real-world data streams necessitates advanced methodologies that can effectively manage the intricacies of these challenges. By examining various approaches in the literature, we aim to highlight their contributions, strengths, and limitations in dealing with imbalanced data distributions, adapting to new class occurrences, and leveraging transfer learning techniques. This comparative analysis not only sheds light on the current state of research but also underscores the specific gaps and unresolved issues that our work seeks to address, ultimately paving the way for more robust and adaptive solutions in the realm of streaming data classification.

3.6.1 Imbalanced Stream

In the context of multi-class classification, addressing class imbalances is crucial. Multi-Label SMOTE (MLSMOTE) extends the principles of SMOTE to generate synthetic examples for each minority class label, thereby enhancing classifier performance by considering neighboring examples in the feature space. Recently, Multi-Label Synthetic Oversampling based on Local label imbalance (MLSOL) has emerged as a more effective technique. MLSOL systematically addresses local imbalances by employing distinct sampling strategies for each label. Empirical research demonstrates that MLSOL outperforms existing methods, including MLSMOTE, in terms of classification accuracy and computational efficiency. By generating synthetic samples exclusively from minority class instances within a restricted neighborhood, MLSOL produces a more compact and efficient synthetic dataset, mitigating concerns related to overfitting. As illustrated in Fig. 3.1, MLSOL is more likely to select $x1$ as a seed instance because it is surrounded by more neighbors of the opposite class for $l3$. MLSMOTE assigns the label vector [0,1,0] to all synthetic instances based on their neighbors. In contrast, MLSOL creates more diverse instances by assigning labels according to their location. Moreover, synthetic instances $c2$ and $c3$ generated by MLSMOTE introduce noise, whereas MLSOL copies the labels of the nearest instance to the new examples. In summary, MLSMOTE tends to generate new instances biased toward the dominant class in the local area, whereas MLSOL effectively explores and exploits both the feature and label space.

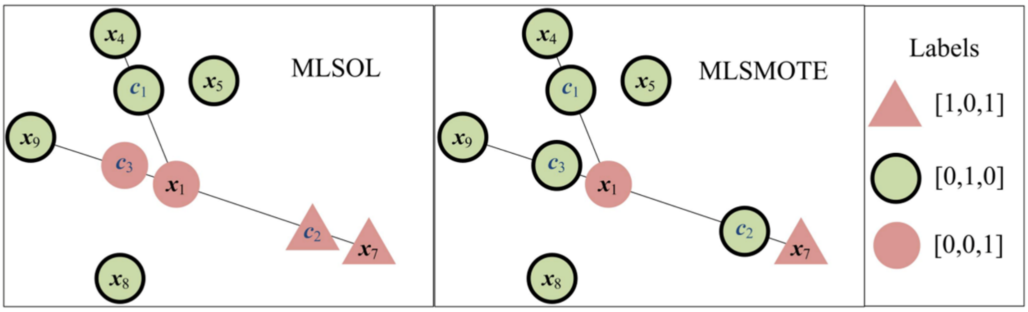


Figure 3.1: Comparsion of MLSMOTE and MLSOL generated instances.
DOI: 110.1016/j.patcog.2021.108294

Table 3.1 presents a comparison of two methods, MLSMOTE and MLSOL, which are designed to address the issue of imbalanced data in multi-class classification. MLSMOTE enhances classifier performance by generating synthetic examples for each minority class label, thereby balancing the class distribution. Its primary advantage is the generation of these synthetic examples, which helps mitigate the imbalance. However, it has significant limitations: the synthetic samples generated might be related to the majority class, which can blur the distinction between classes, and the method struggles with overlapping classes, leading to potential misclassification. On the other hand, MLSOL systematically combats local imbalances by employing distinct sampling strategies for each label within a restricted neighborhood. This localized approach allows MLSOL to generate synthetic examples more precisely, addressing local imbalances effectively. Nonetheless, like MLSMOTE, MLSOL faces challenges with overlapping classes, which can result in misclassification in areas where class boundaries are not clear. Despite its advantages in handling local imbalances, the overlapping class issue remains a critical limitation for both methods, affecting their overall effectiveness in classification tasks.

Table 3.1: Comparison of the MLSMOTE and MLSOL methods.

Method	Theory	Advantages	Limitations
MLSMOTE	MLSMOTE significantly enhances classifier performance by generating synthetic examples for each minority class label.	Generating synthetic examples for each minority class label.	• Random synthetic samples may be related to the majority class. • Overlapping classes.
MLSOL	MLSOL systematically combats local imbalances within the domain of multi-class classification by employing distinct sampling strategies for each label.	Generating synthetic examples for each minority class label within a restricted neighborhood.	• Overlapping classes.

3.6.2 Emergence of new classes

Effectively detecting and adapting to new classes in streaming data is crucial for maintaining classification accuracy. Tree-based methods like SENCForest and SEEN utilize anomaly detection but face high false positive rates and runtime inefficiencies. SENNE improves detection performance using a nearest neighbor ensemble but suffers from longer runtimes due to the lack of an effective model retirement mechanism. The k-nearest Neighbor Ensemble-based method (KNNENS) addresses new class detection and known class classification using a k-nearest neighbor-based hypersphere ensemble and dynamic model updates. However, a critical limitation of these methods is their inadequate handling of concept drift, which is essential for detecting new classes and retraining the classification model. These methods represent the best closely related work for our proposal, which aims to build upon them by incorporating robust concept drift techniques for more adaptive and resilient classification systems.

Fig. 3.2 illustrates the SENCForest approach, which divides the space into three regions (normal, outlying, and anomaly) and detects emerging new classes (anomalies) using a calculated threshold path length. Fig. 3.3 depicts the SENNE algorithm, where hyperplanes are drawn in three dimensions ($x1$, $x2$, and $x3$) for each class (Fig. 3.3a). New instances are then classified as emerging or known classes based on the rank of each class (Fig. 3.3b). Fig. 3.4 presents the KNNENS algorithm, which draws hyperplanes for all class samples (Fig. 3.4a), and classifies new instances using a voting mechanism to determine if the instance is an emerging or known class (Fig. 3.4b). These visualizations highlight the operational differences between the SENNE and KNNENS algorithms in handling the classification of emerging and known classes.

Table3.2 compares three methods for emerging class detection: SENCForest, SENNE, and KNNENS. SENCForest employs the anomaly detection method iForest [20] for new class detection and uses a threshold path to identify anomalies, serving both as an unsupervised anomaly detector and a supervised classifier. However, it has a high potential for false positives and depends on a complex path length

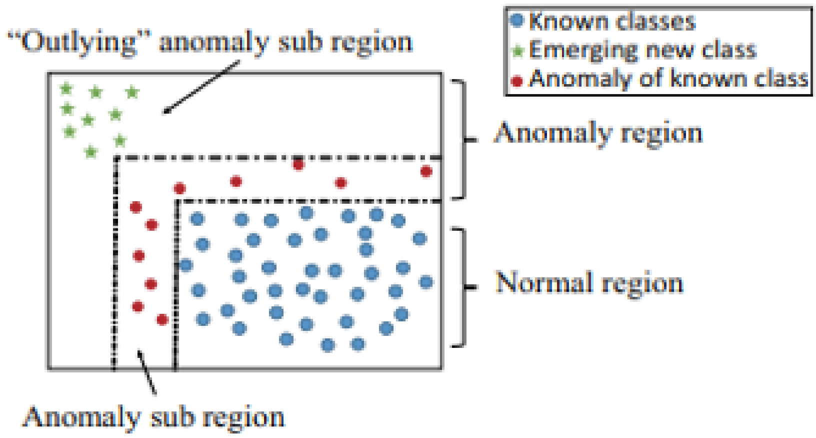
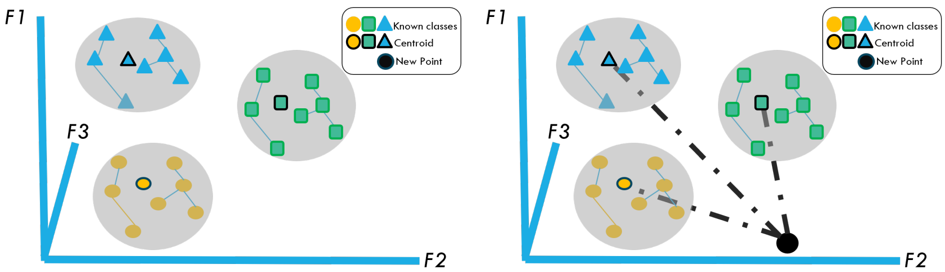


Figure 3.2: Overview of SENCForest detection flow.

DOI: 10.1109/TKDE.2017.2691702

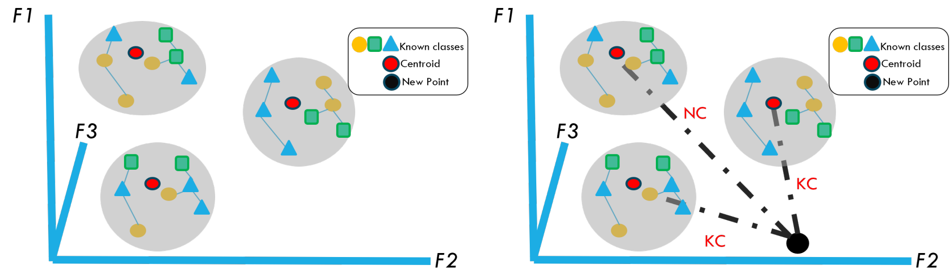


(a)

(b)

Figure 3.3: Overview of the Stream Emerging Nearest Neighbor Ensemble (SENNE).

threshold. SENNE utilizes a nearest neighbor-based hypersphere of one class ensemble to explore local neighborhood information and sort distances, handling both low and high geometric distances between classes. Its limitations include the assumption that the distribution of known classes remains unchanged and it has lengthy update times. KNNENS employs a nearest neighbor-based hypersphere of all class ensembles to explore local neighborhood information, reducing false positives for new classes without needing true labels for model updates. However, like SENNE, it assumes that the distribution of known classes remains unchanged.



(a)

(b)

Figure 3.4: Overview of the k-nearest Neighbor Ensemble-based (KENNE).

Table 3.2: Comparison of the SENCForest, SENNE, and KENNE methods.

Method	Theory	Advantages	Limitations
SENCForst	employs anomaly detection method iForest for a new class detection and then applies threshold path to detect the anomalies.	SENCForest serves as both an unsupervised anomaly detector and a supervised classifier.	• Potential for High false positives. • Dependency on path length threshold (more complexity).
SENNE	nearest neighbor-based hypersphere of one class ensemble to explore local neighborhood information and sort distance to calculate distance.	SENNE is able to handle both the low and high geometric distance between two classes in the feature space.	• Assumes that the distribution of known classes remains unchanged. • Take long time for update.
KENNE	nearest neighbor-based hypersphere of all class ensemble to explore local neighborhood information.	KNNENS to reduce false positives for the new class. KNNENS does not require true labels to update the model.	• Assumes that the distribution of known classes remains unchanged.

3.6.3 Transfer Learning

In the realm of transfer learning, three prominent methods—CORAL, Melanie, and HE-CDTL—serve as closely related approaches to our proposed method. Correlation Alignment (CORAL) is an asymmetric transformation approach that aligns sub-space bases using second-order statistics. By employing a learned transformation matrix, CORAL projects source instances into the target domain, thereby minimizing domain discrepancies and reducing negative knowledge transfer. Melanie addresses the challenge of non-stationary environments through an online ensemble learning approach. It incrementally trains models from both source and target domains, dynamically adjusting their weights to handle concept drift, and combines these models via a weighted-sum approach as shown in Fig. 3.5. As Shown in Fig. 3.6 This method can be extended to Concept Drift Transfer Learning (CDTL) by using an ensemble for chunk-based concept drift. HE-CDTL, designed explicitly for CDTL, leverages knowledge from source domains and historical time steps within the target domain to enhance learning performance. It utilizes a class-wise weighted ensemble for historical knowledge and implements AW-CORAL for extracting knowledge from source domains. The class-wise weighted ensemble allows individual classes to select historical knowledge independently, while AW-CORAL minimizes domain disparities and mitigates negative knowledge transfer. Extensive experiments have shown HE-CDTL to outperform baseline methods in addressing transfer learning challenges in the context of concept drift. Together, these methods provide a comprehensive framework for effective transfer learning in dynamic and evolving data environments.

Table3.3 compares three methods: CORAL, Melanie, and HE-CDTL. CORAL (Correlation Alignment) utilizes a learned transformation matrix and Singular Value Decomposition (SVD) to project source instances into the target domain, effectively minimizing domain discrepancy and reducing negative knowledge transfer. However, it faces challenges with non-stationary and heterogeneous data. Melanie (Multisource Online Transfer learning for Non-stationary Environments) addresses online learning problems where data in source and target domains are generated from non-stationary environments. Its advantages include considering online problems but it too is limited by the complexities of online learning and data heterogeneity. HE-CDTL (Class-wise Weighted and Domain-wise Ensemble) minimizes domain shift by aligning second-order statistics of source and target distributions, leveraging historical knowledge to reduce disparities between domains. Despite its strengths, it relies on the quality of the source domain and also struggles with heterogeneous data.

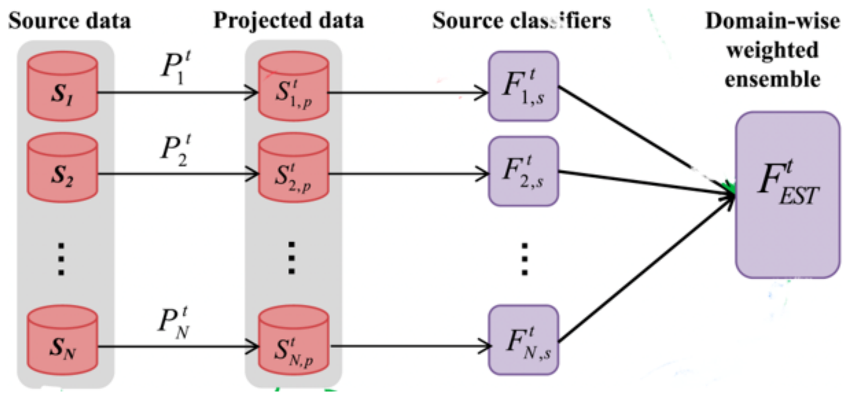


Figure 3.5: Overview of CORrelation ALignment (CORAL)

3.7 Challenges

By comparing the literature on ensemble learning for classification tasks, the proposals in this thesis differ from other studies in several ways:

- As evident from our literature review on imbalanced streams, most studies have concentrated on generating synthetic samples while ignoring class overlap. *To address this challenge*, we propose an approach to generate non-overlapping classes in imbalanced streams.

- Oversampling techniques often perform inefficiently in the presence of concept drift. *To tackle this issue*, we introduce a methodology that selects the oversampling technique based on the current and historical distribution of the stream chunks.

- Our literature review on non-stationary environments reveals that most works focus on detecting emerging new classes while overlooking distribution changes.

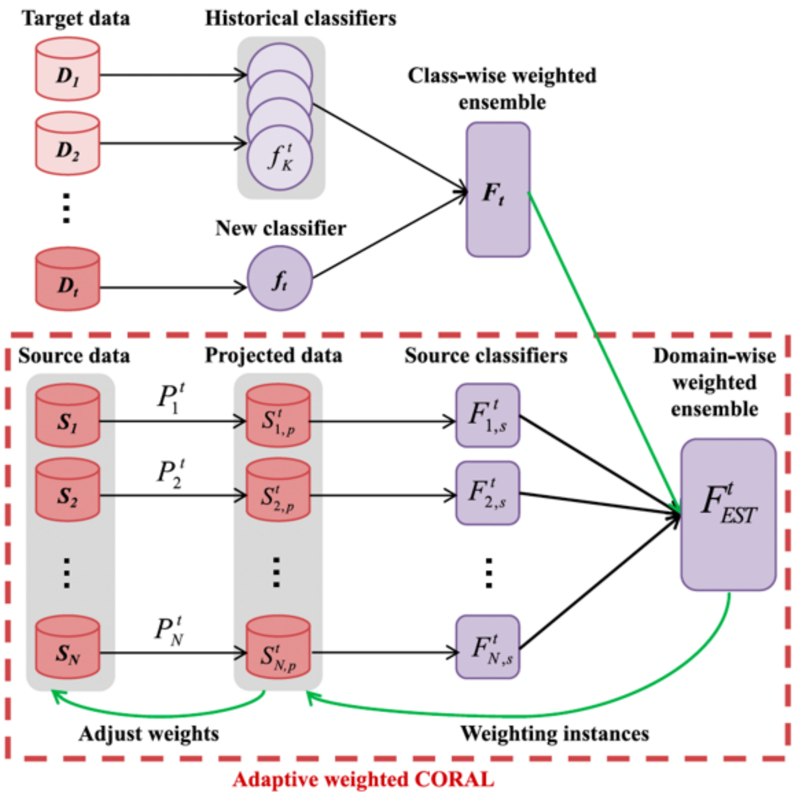


Figure 3.6: Overview of Concept Drift Transfer Learning (CDTL).

To overcome this challenge, we propose a combined approach utilizing Dynamic Ensemble Selection (DES) to select the best classifier for each chunk based on stream distribution, k-means clustering, and concept drift to address both emerging new class detection and distribution changes.

- In our literature review on transfer learning, we observed that most studies focus on homogeneous multisource transfer and neglect heterogeneous multisources in non-stationary environments. *To resolve this issue*, we propose a combined approach integrating Dynamic Ensemble Selection (DES), Concept Drift Transfer Learning (CDTL), eigenvector techniques, and concept drift to address heterogeneous transfer learning in non-stationary environments.

Table 3.3: Comparison of the CORAL, Malanie, and CDTL methods.

Method	Theory	Advantages	Limitations
CORAL	Correlation Alignment (CORAL) uses a learned transformation matrix and Singular Value Decomposition (SVD) to project the source instances into the target domain.	CORAL can minimize domain discrepancy across source and target domains, meanwhile reducing the negative knowledge transfer.	• Non-stationary environments. • Heterogenous multisource.
Melanie	Multi-sourcE onLine TrAnsfer learning for Non-statIonary Environments (Melanie). utilize the class-wise weighted .	It considers an online problem in which the data in source and target domains are generated from non-stationary environments.	• Based on the online learning only. • Heterogenous multisource.
MLSOL	HE-CDTL uses the class-wise weighted and domain wise ensemble for historical knowledge and reduce the disparities between the source and target domains .	HE-CDTL minimizes domain shift by aligning the second-order statistics of source and target distributions.	• Depend on source domain quality. • Heterogenous multisource.

Bibliography

[1] C. Yang, Y.-m. Cheung, J. Ding, and K. C. Tan, "Concept drift-tolerant transfer learning in dynamic environments," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 8, pp. 3857–3871, 2021.

[2] B. Dong, Y. Gao, S. Chandra, and L. Khan, "Multistream classification with relative density ratio estimation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 3478–3485.

[3] J. Shan, H. Zhang, W. Liu, and Q. Liu, "Online active learning ensemble framework for drifted data streams," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 2, pp. 486–498, 2018.

[4] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[5] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.

[6] S. Wang, L. L. Minku, and X. Yao, "A systematic study of online class imbalance learning with concept drift," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 10, pp. 4802–4821, 2018.

[7] Y. Sun, A. K. Wong, and M. S. Kamel, "Classification of imbalanced data: A review," *International journal of pattern recognition and artificial intelligence*, vol. 23, no. 04, pp. 687–719, 2009.

[8] F. Charte, A. J. Rivera, M. J. del Jesus, and F. Herrera, "Addressing imbalance in multilabel classification: Measures and random resampling algorithms," *Neurocomputing*, vol. 163, pp. 3–16, 2015.

[9] ——, "Mlsmote: Approaching imbalanced multilabel learning through synthetic instance generation," *Knowledge-Based Systems*, vol. 89, pp. 385–397, 2015.

[10] Z. Daniels and D. Metaxas, "Addressing imbalance in multi-label classification using structured hellinger forests," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, no. 1, 2017.

[11] B. Liu and G. Tsoumakas, "Making classifier chains resilient to class imbalance," in *Asian Conference on Machine Learning*. PMLR, 2018, pp. 280–295.

[12] U. Bhowan, M. Johnston, M. Zhang, and X. Yao, "Evolving diverse ensembles using genetic programming for classification with unbalanced data," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 3, pp. 368–386, 2012.

[13] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 4, pp. 463–484, 2011.

[14] R. M. Cruz, R. Sabourin, and G. D. Cavalcanti, "Dynamic classifier selection: Recent advances and perspectives," *Information Fusion*, vol. 41, pp. 195–216, 2018.

[15] L. I. Kuncheva, "Clustering-and-selection model for classifier combination," in *KES'2000. Fourth International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies. Proceedings (Cat. No. 00TH8516)*, vol. 1. IEEE, 2000, pp. 185–188.

[16] T. Woloszynski and M. Kurzynski, "A probabilistic model of classifier competence for dynamic ensemble selection," *Pattern Recognition*, vol. 44, no. 10-11, pp. 2656–2668, 2011.

[17] R. Lysiak, M. Kurzynski, and T. Woloszynski, "Optimal selection of ensemble classifiers using measures of competence and diversity of base classifiers," *Neurocomputing*, vol. 126, pp. 29–35, 2014.

[18] R. M. Cruz, R. Sabourin, and G. D. Cavalcanti, "Meta-des. oracle: Meta-learning and feature selection for dynamic ensemble selection," *Information fusion*, vol. 38, pp. 84–103, 2017.

[19] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, "Smoteboost: Improving prediction of the minority class in boosting," in *Knowledge Discovery in Databases: PKDD 2003: 7th European Conference on Principles and Practice of Knowledge Discovery in Databases, Cavtat-Dubrovnik, Croatia, September 22-26, 2003. Proceedings 7.* Springer, 2003, pp. 107–119.

[20] S. Wang, H. Chen, and X. Yao, "Negative correlation learning for classification ensembles," in *The 2010 international joint conference on neural networks (IJCNN).* IEEE, 2010, pp. 1–8.

[21] G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts," *Machine learning*, vol. 23, pp. 69–101, 1996.

[22] N. Lu, J. Lu, G. Zhang, and R. L. De Mantaras, "A concept drift-tolerant case-base editing technique," *Artificial Intelligence*, vol. 230, pp. 108–133, 2016.

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

[24] V. Losing, B. Hammer, and H. Wersing, "Knn classifier with self adjusting memory for heterogeneous concept drift," in *2016 IEEE 16th international conference on data mining (ICDM).* IEEE, 2016, pp. 291–300.

[25] A. Storkey, “When training and test sets are different: characterizing learning transfer,” 2008.

[26] S. Ramírez-Gallego, B. Krawczyk, S. García, M. Woźniak, and F. Herrera, “A survey on data preprocessing for data stream mining: Current status and future directions,” *Neurocomputing*, vol. 239, pp. 39–57, 2017.

[27] J. A. Silva, E. R. Faria, R. C. Barros, E. R. Hruschka, A. C. d. Carvalho, and J. Gama, “Data stream clustering: A survey,” *ACM Computing Surveys (CSUR)*, vol. 46, no. 1, pp. 1–31, 2013.

[28] A. Dries and U. Rückert, “Adaptive concept drift detection,” *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 2, no. 5-6, pp. 311–327, 2009.

[29] C. Alippi and M. Roveri, “Just-in-time adaptive classifiers—part i: Detecting nonstationary changes,” *IEEE Transactions on Neural Networks*, vol. 19, no. 7, pp. 1145–1153, 2008.

[30] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, “Learning with drift detection,” in *Advances in Artificial Intelligence–SBIA 2004: 17th Brazilian Symposium on Artificial Intelligence, Sao Luis, Maranhao, Brazil, September 29-Ocotber 1, 2004. Proceedings 17*. Springer, 2004, pp. 286–295.

[31] L. Bu, C. Alippi, and D. Zhao, “A pdf-free change detection test based on density difference estimation,” *IEEE transactions on neural networks and learning systems*, vol. 29, no. 2, pp. 324–334, 2016.

[32] T. D. S. K. S. Venkatasubramanian and K. Yi, “An information-theoretic approach to detecting changes in multi-dimensional data streams.”

[33] I. Frias-Blanco, J. del Campo-Ávila, G. Ramos-Jimenez, R. Morales-Bueno, A. Ortiz-Diaz, and Y. Caballero-Mota, “Online and non-parametric drift detection methods based on hoeffding’s bounds,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 3, pp. 810–823, 2014.

[34] M. Yamada, A. Kimura, F. Naya, and H. Sawada, "Change-point detection with feature selection in high-dimensional time-series data," in *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

[35] S. H. Bach and M. A. Maloof, "Paired learners for concept drift," in *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 2008, pp. 23–32.

[36] A. Bifet and R. Gavalda, "Learning from time-changing data with adaptive windowing," in *Proceedings of the 2007 SIAM international conference on data mining*. SIAM, 2007, pp. 443–448.

[37] S. Xu and J. Wang, "Dynamic extreme learning machine for data stream classification," *Neurocomputing*, vol. 238, pp. 433–449, 2017.

[38] D. Liu, Y. Wu, and H. Jiang, "Fp-elm: An online sequential learning algorithm for dealing with concept drift," *Neurocomputing*, vol. 207, pp. 322–334, 2016.

[39] D. Han, C. Giraud-Carrier, and S. Li, "Efficient mining of high-speed uncertain data streams," *Applied Intelligence*, vol. 43, pp. 773–785, 2015.

[40] S. G. Soares and R. Araújo, "An adaptive ensemble of on-line extreme learning machines with variable forgetting factor for dynamic system prediction," *Neurocomputing*, vol. 171, pp. 693–707, 2016.

[41] B. F. Manly and D. Mackenzie, "A cumulative sum type of method for environmental monitoring," *Environmetrics: The official journal of the International Environmetrics Society*, vol. 11, no. 2, pp. 151–166, 2000.

[42] Y. Sun, K. Tang, Z. Zhu, and X. Yao, "Concept drift adaptation by exploiting historical knowledge," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 10, pp. 4822–4832, 2018.

[43] N. C. Oza and S. Russell, "Experimental comparisons of online and batch versions of bagging and boosting," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, 2001, pp. 359–364.

[44] A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, and R. Gavalda, "New ensemble methods for evolving data streams," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, pp. 139–148.

[45] F. Chu and C. Zaniolo, "Fast and light boosting for adaptive mining of data streams," in *Advances in Knowledge Discovery and Data Mining: 8th Pacific-Asia Conference, PAKDD 2004, Sydney, Australia, May 26-28, 2004. Proceedings 8*. Springer, 2004, pp. 282–292.

[46] H. M. Gomes, A. Bifet, J. Read, J. P. Barddal, F. Enembreck, B. Pfharinger, G. Holmes, and T. Abdessalem, "Adaptive random forests for evolving data stream classification," *Machine Learning*, vol. 106, pp. 1469–1495, 2017.

[47] P. Li, X. Wu, X. Hu, and H. Wang, "Learning concept-drifting data streams with random ensemble decision trees," *Neurocomputing*, vol. 166, pp. 68–83, 2015.

[48] J. Z. Kolter and M. A. Maloof, "Dynamic weighted majority: An ensemble method for drifting concepts," *The Journal of Machine Learning Research*, vol. 8, pp. 2755–2790, 2007.

[49] R. Elwell and R. Polikar, "Incremental learning of concept drift in nonstationary environments," *IEEE transactions on neural networks*, vol. 22, no. 10, pp. 1517–1531, 2011.

[50] D. Brzezinski and J. Stefanowski, "Reacting to different types of concept drift: The accuracy updated ensemble algorithm," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 1, pp. 81–94, 2013.

[51] P. Zhang, X. Zhu, and Y. Shi, "Categorizing and mining concept drifting data streams," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, pp. 812–820.

[52] Y. Sun, K. Tang, L. L. Minku, S. Wang, and X. Yao, "Online ensemble learning of data streams with gradually evolved classes," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 6, pp. 1532–1545, 2016.

[53] J. B. Gomes, M. M. Gaber, P. A. Sousa, and E. Menasalvas, "Mining recurring concepts in a dynamic feature space," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 1, pp. 95–110, 2013.

[54] Z. Ahmadi and S. Kramer, "Modeling recurring concepts in data streams: a graph-based framework," *Knowledge and Information Systems*, vol. 55, pp. 15–44, 2018.

[55] M. Pratama, J. Lu, and G. Zhang, "Evolving type-2 fuzzy classifier," *IEEE Transactions on Fuzzy Systems*, vol. 24, no. 3, pp. 574–589, 2015.

[56] P. Domingos and G. Hulten, "Mining high-speed data streams," in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2000, pp. 71–80.

[57] G. Hulten, L. Spencer, and P. Domingos, "Mining time-changing data streams," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, 2001, pp. 97–106.

[58] H. Yang and S. Fong, "Incrementally optimized decision tree for noisy big data," in *Proceedings of the 1st International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications*, 2012, pp. 36–44.

[59] ——, "Countering the concept-drift problems in big data by an incrementally optimized stream mining model," *Journal of Systems and Software*, vol. 102, pp. 158–166, 2015.

[60] L. Rutkowski, L. Pietruczuk, P. Duda, and M. Jaworski, "Decision trees for mining data streams based on the mcdiarmid's bound," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 6, pp. 1272–1279, 2012.

[61] L. Rutkowski, M. Jaworski, L. Pietruczuk, and P. Duda, "A new method for data stream mining based on the misclassification error," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 5, pp. 1048–1059, 2014.

[62] I. Frias-Blanco, J. del Campo-Avila, G. Ramos-Jimenez, A. C. Carvalho, A. Ortiz-Díaz, and R. Morales-Bueno, "Online adaptive decision trees based on concentration inequalities," *Knowledge-Based Systems*, vol. 104, pp. 179–194, 2016.

[63] M. Baena-Garcıa, J. del Campo-Ávila, R. Fidalgo, A. Bifet, R. Gavalda, and R. Morales-Bueno, "Early drift detection method," in *Fourth international workshop on knowledge discovery from data streams*, vol. 6. Citeseer, 2006, pp. 77–86.

[64] A. H. Madkour, A. Elsayed, and H. Abdel-Kader, "Historical isolated forest for detecting and adaptation concept drifts in nonstationary data streaming," *IJCI. International Journal of Computers and Information*, vol. 10, no. 2, pp. 16–27, 2023.

[65] C. H. Tan, V. C. Lee, and M. Salehi, "Information resources estimation for accurate distribution-based concept drift detection," *Information Processing & Management*, vol. 59, no. 3, p. 102911, 2022.

[66] J. N. Adams, S. J. van Zelst, T. Rose, and W. M. van der Aalst, "Explainable concept drift in process mining," *Information Systems*, vol. 114, p. 102177, 2023.

[67] E. S. Page, "Continuous inspection schemes," *Biometrika*, vol. 41, no. 1/2, pp. 100–115, 1954.

[68] K. Jackowski, B. Krawczyk, and M. Woźniak, "Improved adaptive splitting and selection: the hybrid training method of a classifier based on a feature space partitioning," *International journal of neural systems*, vol. 24, no. 03, p. 1430007, 2014.

[69] T. Yin, C. Liu, F. Ding, Z. Feng, B. Yuan, and N. Zhang, "Graph-based stock correlation and prediction for high-frequency trading systems," *Pattern Recognition*, vol. 122, p. 108209, 2022.

[70] J. Ren, Y. Wang, Y.-m. Cheung, X.-Z. Gao, and X. Guo, "Grouping-based oversampling in kernel space for imbalanced data classification," *Pattern Recognition*, vol. 133, p. 108992, 2023.

[71] V. C. Nitesh, "Smote: synthetic minority over-sampling technique," *J Artif Intell Res*, vol. 16, no. 1, p. 321, 2002.

[72] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-smote: a new over-sampling method in imbalanced data sets learning," in *International conference on intelligent computing*. Springer, 2005, pp. 878–887.

[73] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, "Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem," in *Advances in knowledge discovery and data mining: 13th Pacific-Asia conference, PAKDD 2009 Bangkok, Thailand, April 27-30, 2009 proceedings 13*. Springer, 2009, pp. 475–482.

[74] T. Maciejewski and J. Stefanowski, "Local neighbourhood extension of smote for mining imbalanced data," in *2011 IEEE symposium on computational intelligence and data mining (CIDM)*. IEEE, 2011, pp. 104–111.

[75] Y. Gao, S. Chandra, Y. Li, L. Khan, and T. Bhavani, "Saccos: A semi-supervised framework for emerging class detection and concept drift adaption over data streams," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 3, pp. 1416–1426, 2020.

[76] M. Masud, J. Gao, L. Khan, J. Han, and B. M. Thuraisingham, "Classification and novel class detection in concept-drifting data streams under time constraints," *IEEE Transactions on knowledge and data engineering*, vol. 23, no. 6, pp. 859–874, 2010.

[77] A. Haque, L. Khan, and M. Baron, "Sand: Semi-supervised adaptive novel class detection and classification over data stream," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[78] X. Mu, F. Zhu, J. Du, E.-P. Lim, and Z.-H. Zhou, “Streaming classification with emerging new class by class matrix sketching,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.

[79] X. Mu, K. M. Ting, and Z.-H. Zhou, “Classification under streaming emerging new classes: A solution using completely-random trees,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 8, pp. 1605–1618, 2017.

[80] Y.-N. Zhu and Y.-F. Li, “Semi-supervised streaming learning with emerging new labels,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 7015–7022.

[81] X.-Q. Cai, P. Zhao, K.-M. Ting, X. Mu, and Y. Jiang, “Nearest neighbor ensembles: An effective method for difficult problems in streaming classification with emerging new classes,” in *2019 IEEE international conference on data mining (ICDM)*. IEEE, 2019, pp. 970–975.

[82] J. Zhang, T. Wang, W. W. Ng, and W. Pedrycz, “Knnens: A k-nearest neighbor ensemble-based method for incremental learning under data stream with emerging new classes,” *IEEE transactions on neural networks and learning systems*, vol. 34, no. 11, pp. 9520–9527, 2022.

[83] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, “Transfer feature learning with joint distribution adaptation,” in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 2200–2207.

[84] ——, “Transfer joint matching for unsupervised domain adaptation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1410–1417.

[85] Q. Sun, R. Chattopadhyay, S. Panchanathan, and J. Ye, “A two-stage weighting framework for multi-source domain adaptation,” *Advances in neural information processing systems*, vol. 24, 2011.

[86] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Boosting for transfer learning," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 193–200.

[87] Y. Freund, R. E. Schapire *et al.*, "Experiments with a new boosting algorithm," in *icml*, vol. 96. Citeseer, 1996, pp. 148–156.

[88] Y. Yao and G. Doretto, "Boosting for transfer learning with multiple sources," in *2010 IEEE computer society conference on computer vision and pattern recognition*. IEEE, 2010, pp. 1855–1862.

[89] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, no. 1, 2016.

[90] Y. Zhu, F. Zhuang, J. Wang, G. Ke, J. Chen, J. Bian, H. Xiong, and Q. He, "Deep subdomain adaptation network for image classification," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 4, pp. 1713–1722, 2020.

[91] M. Long, Y. Cao, Z. Cao, J. Wang, and M. I. Jordan, "Transferable representation learning with deep adaptation networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 12, pp. 3071–3085, 2018.

[92] J. Wang, Y. Chen, W. Feng, H. Yu, M. Huang, and Q. Yang, "Transfer learning with dynamic distribution adaptation," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 11, no. 1, pp. 1–25, 2020.

[93] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 2960–2967.

[94] K. Pearson, "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, vol. 2, no. 11, pp. 559–572, 1901.

[95] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE transactions on neural networks*, vol. 22, no. 2, pp. 199–210, 2010.

[96] M. M. Rahman, C. Fookes, M. Baktashmotlagh, and S. Sridharan, "Correlation-aware adversarial domain adaptation and generalization," *Pattern Recognition*, vol. 100, p. 107124, 2020.

[97] E. Zhong, W. Fan, J. Peng, K. Zhang, J. Ren, D. Turaga, and O. Verscheure, "Cross domain distribution adaptation via kernel mapping," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, pp. 1027–1036.

[98] Z. Cao, M. Long, J. Wang, and M. I. Jordan, "Partial transfer learning with selective adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2724–2732.

[99] Z. Cao, K. You, M. Long, J. Wang, and Q. Yang, "Learning to transfer examples for partial domain adaptation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 2985–2994.

[100] L. Li, Z. Wan, and H. He, "Dual alignment for partial domain adaptation," *IEEE transactions on cybernetics*, vol. 51, no. 7, pp. 3404–3416, 2020.

[101] D. M. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," *arXiv preprint arXiv:2010.16061*, 2020.