Analysis of the impact of Dissimilarity Space within the Concept Drift Problem

1st Pedro Bianchini de Quadros
Polytechnic School
Pontifícia Universidade Católica do Paraná (PUCPR)
Curitiba, Brazil
quadros.pedro@pucpr.edu.br

2nd Gabriel Antonio Gomes de Farias
Polytechnic School
Pontifícia Universidade Católica do Paraná (PUCPR)
Curitiba, Brazil
gomes.farias@pucpr.edu.br

3rd Jean Paul Barddal
Graduate Program in Informatics (PPGIa)
Pontifícia Universidade Católica do Paraná (PUCPR)
Curitiba, Brazil
jean.barddal@ppgia.pucpr.br

***Abstract*—Concept Drift is a known problem mostly in the context of supervised models in which it affects negatively many models capabilities due to the change of the original concept and thus generating wrong target values. Due to time or other phenomena, i.e., changes over time during many possible phases in such a way that it deviates from generating the original intended target because of those changes for the model to capture. Researchers are continuously trying to find solutions that may be able to identify concept drift and also to quickly adapt predictive models so that its impact is reduced or entirely mitigated. In this paper, our goal is to apply classification techniques based on dissimilarity space representation based on various literature in hopes that it may solve most of the divergence caused by the problem and thus making a robust model in which can handle the problem at hand and thus diminishing it's impact in such that the model targets maintain the correct classification even due to changes overtime. The results indicate that the transformation of data into the dissimilarity space did not yield significant benefits. One of the classifiers utilizing Instance Hardness metrics demonstrated performance comparable to that of traditional classifiers.**

***Index Terms*—Machine Learning, Dissimilarity spaces, Concept drift, Data streams, Data Science.**

I. Introduction

In modern times, the internet manages a plethora of data streams, whether from a large e-commerce conglomerate, a social network handling millions of user data points, or a simple artisan's blog. These data, as shown by Babcock [1], are obtained via information traffic and then processed within specific applications, each with a particular purpose, such as labeling, which is prevalent in most classification problems. There are applications where data streams are of great importance, as the data must be received and processed continuously, and in some cases, in real time.

An example is an environment with multiple sensors that read a particular phenomenon to label various conditions, such as "humid" or "dry." If one of these sensors is altered due to aging, calibration shifts, or replacement with a newer model—there may be changes in the collected data that directly impact the application, generating discrepancies and anomalies in classification tasks due to data inconsistency.

This is just one scenario, but many real-world applications, such as those in health, geography, finance, and other fields, can be affected and suffer significant consequences. Therefore, solutions are needed to address this issue.

The data that constitute a system are essential for the operation of many applications and systems. However, cases known as concept drift can lead to significant drawbacks at various levels for these applications [2]. Concept drift is, in essence, a shift in data recognition, commonly encountered in classification problems within the realm of machine learning. This drift can result from various factors, such as time, which degrades the system, sensors, and components, the replacement of a part or even an entire application, or the evolution of classes, e.g., the addition of new classes, in a classification problem. These alterations directly impact classification tasks, giving rise to the concept drift problem.

In the examples provided, it is evident that this issue affects various fields, with impacts ranging from mild to severe, as in the case of pathology detection in medicine or the analysis of geographic and biological phenomena to predict possible natural calamities.

There are contingency methods to address cases of concept drift. However, an important step would be to detect the concept drift itself, and for this, algorithms such as the Drift Detection Method, proposed by Gama et al. [3], and ADWIN [4] are available. These algorithms focus on identifying statistical discrepancies over time within a dataset. Algorithms and detectors of this type are crucial for identifying when concept drift occurs, serving as an essential tool for understanding our data.

The development of solutions that mitigate issues caused by these drifts brings significant benefits to all sectors of society dealing with computational classification tasks, primarily to achieve more robust and reliable models capable of handling these drifts while remaining operationally optimal, thus conserving resources across various domains.

An essential concept for this research is that of dissimilarity space, which refers to a proposal based on a type of classification grounded in the concept of dissimilarity. This approach was proposed by E. Pekalska et al. [5], where classification models can be learned after converting data from its original attribute space to the dissimilarity attribute space.

The available literature suggests the potential utility of such a classifier due to its ability to operate independently of patterns. In particular, traditional classification models rely on commonly used pattern recognition techniques, whereas dissimilarity recognition does not depend on this since it does not operate by classifying in the way that we humans are accustomed to. In potential applications, a classifier based on this concept may be important for addressing problems affected by concept drift, as it seeks to measure the degree of dissimilarity.

The hypothesis of our work is as follows:

1) H0: The representation of a dataset in a dissimilarity space results in lower dispersion of the data in concept drift scenarios.
2) H1: The representation of a dataset in a dissimilarity space results in equal or greater dispersion in concept drift scenarios.

This paper is divided as follows. Section II introduces and formalized concept drift, which is the main target of our investigation. Section III describes our proposal, in which we explore the use of the dissimilarity space as an alternative for data classification under concept drift. In Section IV, we detail the methodology for testing the proposed algorithms. This section outlines the experimental design, including the datasets that will be utilized, the performance metrics for evaluation, and the statistical methods employed to analyze the results. By conducting a series of experiments, we aim to assess the effectiveness of our proposed approach in addressing the challenges posed by concept drift. We will compare the performance of classifiers operating in traditional attribute spaces against those functioning in the dissimilarity space to determine the advantages and limitations of each method. We conclude the paper by discussing the implications of our findings, potential areas for further research, and the broader impact of our work on the field of computational classification.

II. Concept Drift

.

Concept drift is when you a model trained and due to something changing over time or due to other phenomena, such as shifts, or calibration changes, these changes cause the model's predictions to deviate from expected behavior, leading to inaccurate target values. This happens because the original concept the model was built upon has been altered, which can result in a decline in model performance or even render it obsolete.

According to Gama et al. [6], it can be mathematically defined as:

$$\exists X : p_{t_0}(X, y) \neq p_{t_1}(X, y)$$

Where X represents the set of features (or attributes) that describe the instances in the dataset, and y is the target variable, meaning what we are trying to predict or classify.

The notations $p_{t_0}(X, y)$ and $p_{t_1}(X, y)$ represent, respectively, the joint probability distributions of the features X and the variable y at two different points in time, t_0 and t_1.

The notation $\exists X$ indicates that there exists at least one instance where the probability distribution of y, conditioned on X, changes over time. This implies the occurrence of "concept drift," meaning that the relationship between the features and the target variable has altered over time.

In addition to this definition, drifts are often categorized into real concept drift and virtual concept drift. Real concept drift refers to changes in $p(y|X)$, which is the probability of observing each class given the input attributes of the observed class. These changes can occur with or without changes in the input data $p(X)$. According to the review by Gama et al. [6], this concept was guided by Salganicoff (1997) in the article "Tolerating Concept and Sampling Shift in Lazy Learning Using Prediction Error Context Switching" [7] and referred to as conditional change by Gao et al. (2007) in the article "A General Framework for Mining Concept-Drifting Data Streams with Skewed Distributions" [8] presented at the 7th SIAM International Conference on Data Mining.

For virtual concept drift, according to the same review, it occurs when the distribution of the received data changes, meaning that $p(X)$ changes without affecting $p(y|X)$. According to the review, this concept is presented by Delany et al. in the article "A Case-Based Technique for Tracking Concept Drift in Spam Filtering" [9] as well as by Tsymbal in "The Problem of Concept Drift: Definitions and Related Work. Technical Report, Department of Computer Science" [10] and by Widmer and Kubat in 1993 in the article "Effective Learning in Dynamic Environments by Explicit Context Tracking." It is also important to note that the definition of virtual concept drift has several different interpretations in the literature however, for this research project, we will only analyze problems involving real concept drift.

III. Proposal

To address the problem of concept drift, we propose the use of the dissimilarity space. According to the article "The Dissimilarity Approach: A Review" [11], the dissimilarity space is discussed in the context of classification problems where patterns have an intrinsic and detectable organization, as seen with shapes, images, or texts. These patterns exhibit latent aspects, such as order, time, hierarchy, or functional relationships.

In the same study, two widely used strategies are presented: the dissimilarity space and the dissimilarity vector. For this study, we will focus specifically on the dissimilarity space, which consists of a matrix that represents the disparity between the analyzed points, reflecting their similarity or difference.

Upon reviewing the existing literature, several approaches to dealing with concept drift can be noted, both in online and offline settings. The article presents incremental algorithms, which are offline approaches where examples are added sequentially (or in batches) to the data, and after this addition, the model is updated.

Other approaches for detecting concept drift have also been identified, in which dissimilarity is used to detect these changes. In the articles “A Dissimilarity-based Drift Detection Method” [12] and “Experimental Analysis on Dissimilarity Metrics and Sudden Concept Drift Detection” [13], we encounter this alternative approach.

In the work by Pinagé and Dos Santos, a new approach is presented where, in the incremental algorithm, the model is updated with each new example received. However, as discussed in the article, this approach may lead to unnecessary updates in the system. To address this issue, the authors developed a method that detects changes based on dissimilarity in the data distribution, updating the decision model only when a change is detected. The experiments conducted in this research showed that the detection rates achieved were comparable to traditional concept drift detectors, such as DDM and EDDM, which are based on incremental learning and performance monitoring.

In the work by Basterrech et al., similar to the study by Pinagé and Dos Santos, dissimilarity is used to detect concept drift. However, this article employs a dissimilarity metric weighted on posterior probabilities, thus achieving significant results.

According to these studies, we will analyze the impact of dissimilarity on the concept drift problem, aiming to evaluate how this approach can improve model adaptation in the presence of concept changes in the data.

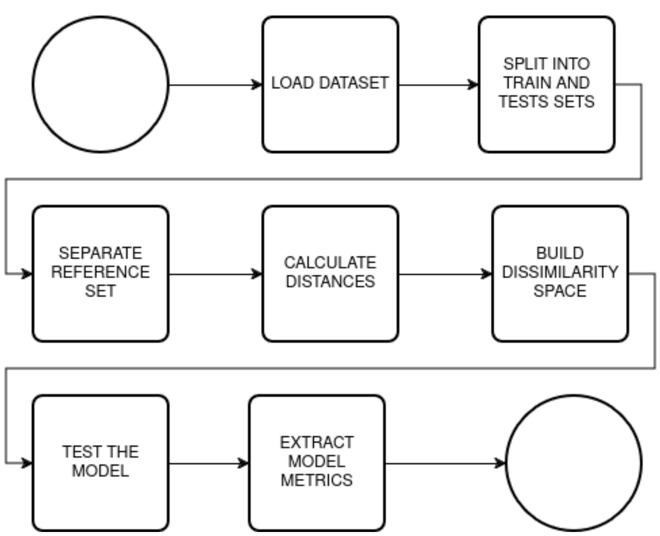

Fig. 1. Classifer flowchart.

As shown in Figure 1, the algorithm begins by splitting the data into two subsets: training and testing. We denote the training set as $T = \{x_1, x_2, \ldots, x_n\}$ and the testing set as $T' = \{z_1, z_2, \ldots, z_m\}$.

After the initial split, we proceed to calculate the dissimilarity space between the instances of the training set. For this, we select a reference set $R = p_1, p_2, \ldots, p_m$, which will be used as the basis for calculating the distances. The dissimilarity between the instances in the training set T and the reference set R is obtained through the Euclidean distance, given by the following equation:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

This calculation results in a dissimilarity matrix $\boldsymbol{D}$, also called a dissimilarity space, which contains the distances between the instances of T and R, representing the dissimilarity between them. The dissimilarity space is, therefore, constructed based on this distance matrix.

A relevant aspect of this process is the selection of instances that will compose the reference set R. We proceeded with analyzing the following approaches:

1) **Random Selection**
   In this approach, we will use instances from the training set T that are selected randomly to form the reference set R.
   Let T be a set of instances. We want to form a subset R such that $R \subseteq T$ and $|R| = x$, where x is the number of randomly selected instances. Mathematically, we can represent this as follows:

$$R \subseteq T \quad \text{and} \quad |R| = x$$

   Thus, R is a set formed by x elements of T:

$$R = \{t_1, t_2, \ldots, t_x \mid t_i \in T\}$$

2) **K-means**

   In this approach, we use the K-means algorithm to segment the training set T into k clusters. The algorithm organizes the data into groups, and for each cluster, it selects a centroid that serves as a representation of the group. These centroids are then used to form a new set, R, which is a subset of the training set T.
   Formally, we can define $R \subseteq T$, where $|R| = x$, with x being the number of selected centroids. Thus, R consists of x elements extracted from T, that is:

$$R = \{t_1, t_2, \ldots, t_x \mid t_i \in T\}$$

   Each element t_i in R represents a centroid associated with one of the clusters formed in the set T.

3) **Instance Hardness**

   For Instance Hardness, we calculate the instance hardness. In Smith’s paper [14], it is shown that instance hardness is the property indicating the likelihood of an instance being misclassified. In other words, outliers and incorrectly labeled instances are expected to have high instance hardness, as a learning algorithm will need to overfit to classify them correctly.

Knowing this, the same paper presents metrics that evaluate various aspects of the hardness level of an individual instance. For this research, we will use the kDN, which measures the local overlap of an instance in the original task space relative to its nearest neighbors (using Euclidean distance) that do not share the same target class value.

$$\frac{k\text{DN}(x) = |\{y : y \in k\text{NN}(x) \wedge t(y) \neq t(x)\}|}{k}$$

The $k\text{NN}(x)$ is the set of k nearest neighbors of x, and $t(x)$ is the target class for x. From this metric, we select the instances from the set T that have the highest hardness values, thus forming the set R. Formally, we can define $R \subseteq T$, where $|R| = x$, with x being the number of selected instances. Thus, R consists of x elements extracted from T:

$$R = \{t_1, t_2, \ldots, t_x \mid t_i \in T\}$$

Each element t_i in R represents a selected instance with high hardness, ensuring that R captures the most challenging characteristics of the set T. This selection process is crucial for enhancing the efficiency of our model, as it focuses the analysis on the instances that most impact the classifier's performance.

After selecting set R and constructing the dissimilarity space, we use the set T' to test the model. In this research, we are testing with data that presents concept drift, which we will explain in more detail in Section IV, where we provide further details on how the models were tested. We also collected the metrics for the tested models and approaches.

IV. Analysis

In this section, we present the experimental protocol adopted to assess our proposal, followed by an analysis of the results obtained. The experimental protocol outlines the steps taken to simulate concept drift using a dataset generated by MOA, detailing the structure of the experiments conducted. Specifically, we describe how instances from the training set are randomly selected to form a reference set, and we delineate the classifiers employed, including traditional and proposed models. Finally, we analyze the results of the experiments, providing organized tables for visualization and interpretation of the classifier performances.

A. Experimental Protocol

To simulate concept drift and analyze the results, we will perform two experiments. We will generate a dataset using MOA [15] with 100,000 instances and apply concept drift starting from instance 50,000 onwards. Thus, the dataset will have an initial concept and an altered concept, with a training and testing split of 50/50.

The two experiments will be as follows:

1) Train classifiers with the initial concept and altered concept, in a 50/50 proportion.
2) Divided into two stages:
   a) From the first half of the dataset (initial concept), 25% will be used for training and 25% for testing.
   b) Using 25% of the first half (initial concept) for training and the entire second half (altered concept) for testing.

In this study, we employed traditional classifiers such as K-Nearest Neighbors, considering both the version with the nearest neighbor (KNN1) and the version with the three nearest neighbors (KNN3), as well as the Gaussian Naive Bayes and Decision Tree classifiers. These were used alongside the proposed classifiers: DissimilarityRNGClassifier, DissimilarityCentroidClassifier, and DissimilarityIHDClassifier, which were developed to explore different approaches to constructing the dissimilarity space. To perform the experiments, we used synthetic datasets generated by MOA, which include:

TABLE I
DATASET CHARACTERISTICS

Dataset	Total Instances	Attributes	Classes
AgrawalGenerator [16]	100,000	9	2
AssetNegotiationGenerator [17]	100,000	5	2
SEAGenerator [18]	100,000	3	2

Finally, to evaluate the classifiers applied to this problem, we use accuracy as the performance metric. The accuracy is mathematically defined as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives, and FN is the number of false negatives.

B. Analysis

The experiments were conducted following the protocol described in Section IV-A, with results organized into tables for ease of visualization. In each table, we grouped classifiers, the applied estimators, and their respective accuracies, enabling direct comparisons between the strategies and techniques employed. In order to facilitate the interpretation of all tables, we provide a legend of abbreviations for the classifiers (see Table II)

TABLE II
TABLE OF ABBREVIATIONS AND CLASSIFIERS

Abbreviation	Classifier
DTC	DecisionTreeClassifier
DIHD	DissimilarityIHDClassifier
DC	DissimilarityCentroidClassifier
DRNG	DissimilarityRNGClassifier
GNB	GaussianNB
KNN1	KNeighborsClassifier - Neighbors 1
KNN3	KNeighborsClassifier - Neighbors 3

1) Test 1: Tables III and IV display the accuracies of the classifiers tested with two approaches: the "*per class*" and "*all class*" strategies. It is observed that the Dissimilarity Instance Hardness(DIHD), based on instance hardness, showed

TABLE III
AVERAGE OF CLASSIFIERS USING THE PER CLASS STRATEGY

Classifier	Estimator	Accuracy
GNB	Empty estimator	69.315
DIHD	KNN3	66.078
KNN3	Empty estimator	65.599
DRNG	KNN3	65.159
DIHD	DT	64.417
DIHD	KNN1	64.123
DC	KNN3	63.954
KNN1	Empty estimator	63.733
DT	Empty estimator	63.687
DC	KNN1	62.701
DRNG	KNN1	62.567
DC	DT	61.727
DC	GNB	61.500
DRNG	DT	61.361
DIHD	GNB	56.595
DRNG	GNB	56.501

TABLE IV
AVERAGE OF CLASSIFIERS USING THE ALL CLASS STRATEGY

Classifier	Estimator	Accuracy
GNB	Empty estimator	69.315
KNN3	Empty estimator	65.599
DIHD	KNN3	65.273
DRNG	KNN3	65.159
KNN1	Empty estimator	63.733
DT	Empty estimator	63.687
DIHD	KNN1	63.173
DIHD	DT	63.035
DC	KNN3	62.663
DRNG	KNN1	62.567
DRNG	DT	61.361
DC	DT	61.254
DC	KNN1	61.223
DC	GNB	58.328
DRNG	GNB	56.501
DIHD	GNB	51.041

TABLE V
AVERAGE OF CLASSIFIERS USING THE PER CLASS STRATEGY, METHOD 1

Classifier	Estimator	Accuracy
DT	Empty estimator	88.011
DIHD	KNN3	81.992
DC	KNN3	81.610
DRNG	KNN3	81.035
GNB	Empty estimator	80.353
KNN3	Empty estimator	80.237
DRNG	DT	78.224
DC	DT	77.940
DIHD	DT	77.916
DIHD	KNN1	77.879
DC	KNN1	77.750
DRNG	KNN1	77.083
KNN1	Empty estimator	76.764
DC	GNB	75.727
DIHD	GNB	74.865
DRNG	GNB	72.197

TABLE VI
AVERAGE OF CLASSIFIERS USING THE ALL CLASS STRATEGY, METHOD 1

Classifier	Estimator	Accuracy
DT	Empty estimator	88.011
DRNG	KNN3	81.035
GNB	Empty estimator	80.353
KNN3	Empty estimator	80.237
DIHD	KNN3	78.229
DRNG	DT	78.224
DRNG	KNN1	77.083
KNN1	Empty estimator	76.764
DC	KNN3	76.275
DIHD	DT	76.150
DIHD	KNN1	74.336
DC	DT	73.128
DC	KNN1	72.552
DRNG	GNB	72.197
DC	GNB	71.641
DIHD	GNB	67.076

slightly better performance than the other proposed classifiers, suggesting an advantage in more complex classifications, though without statistical significance. This performance may be attributed to DIHD's focus on harder instances, which could aid in imbalanced classes, although the improvement observed remains suable.

2) Test 2: During the initial phase of testing, which involved training the classifiers using only the preliminary concept, the results fell short of expectations. The average accuracies ohbained were either lower than or equivalent to those achieved with established traditional classifiers, indicating that the new initial approach did not yield significant improvements compared to conventional methods. Tables V and VI present the outcomes of the two strategies employed: an overview of the classifiers and estimators applied, along with their average accuracies. Notably, the Decision Tree (DT) classifier demonstrated the highest accuracy in both tables, whereas other classifiers, such as Dissimilarity Random Number Generator (DRNG) and Dissimilarity Instance Hardness (DIHD), achieved lower accuracies, particularly when combined with the Gaussian Naive Bayes (GNB) estimator.

In the second stage of testing, 25% of the first half (initial concept) was utilized for training, while the entirety of the second half (altered concept) was reserved for testing. The results remained consistent with those of the first stage, showing no significant improvements in the average accuracy of the classifiers. Tables VII and VIII present the performance of the classifiers under the two evaluation strategies.

Table VII summarizes the results of the per class strategy, indicating that the Decision Tree (DT) classifier with an empty estimator continues to exhibit the highest average accuracy (75.596%), while other methods, such as Gaussian Naive Bayes (GNB) and Dissimilarity Instance Hardness (DIHD) with various estimators, demonstrated more varied performance, reinforcing the limited stability of these classifiers for the altered concept.

Conversely, Table VIII, which pertains to the all class strategy, reveals that the DT maintains its lead, achieving results slightly superior to those of the other classifiers, followed by GNB. These findings suggest that, despite the variability among classifiers and estimators, the introduction of the altered concept did not positively impact accuracy, indicating a need

TABLE VII
AVERAGE OF CLASSIFIERS USING THE PER CLASS STRATEGY, METHOD 2

Classifier	Estimator	Accuracy
DT	Empty estimator	75.596
GNB	Empty estimator	74.596
DIHD	KNN3	74.096
DRNG	KNN3	72.824
DC	KNN3	72.442
KNN3	Empty estimator	72.400
DIHD	KNN1	70.492
DRNG	KNN1	69.769
KNN1	Empty estimator	69.744
DIHD	DT	69.713
DRNG	DT	69.514
DC	KNN1	69.302
DC	DT	69.129
DC	GNB	68.462
DIHD	GNB	66.520
DRNG	GNB	64.722

TABLE VIII
AVERAGE OF CLASSIFIERS USING THE ALL CLASS STRATEGY, METHOD 2

Classifier	Estimator	Accuracy
DT	Empty estimator	75.596
GNB	Empty estimator	74.596
DRNG	KNN3	72.824
DIHD	KNN3	72.798
KNN3	Empty estimator	72.400
DIHD	DT	70.149
DRNG	KNN1	69.769
KNN1	Empty estimator	69.744
DIHD	KNN1	69.605
DC	KNN3	69.566
DRNG	DT	69.514
DC	DT	66.591
DC	KNN1	66.435
DC	GNB	65.051
DRNG	GNB	64.722
DIHD	GNB	63.162

for novel approaches to enhance generalization.

V. CONCLUSION

The challenge associated with the concept drift involved investigating how modifications in the dissimilarity space impact classifier accuracy. Our goal was to demonstrate the effect of this change on model performance, highlighting the implications that the structure of the dissimilarity space may have on the adaptability and accuracy of classifiers when confronted with varying concepts.

To address this challenge, we conducted tests with existing classifiers and implemented new models that utilize the dissimilarity space during training. This approach aims to enhance classification efficacy by enabling more efficient adaptation to changes in the data.

The results indicate that the new classifiers, by incorporating the dissimilarity space into their training process, showed performance that was comparable to or lower than traditional methods. According to the initial hypothesis, we were unable to demonstrate that this was a good strategy, as there was no significant increase in accuracy. This confirms hypothesis H1, which states that the dispersion of the data is equal to or greater than that of traditional classifiers.

However, when analyzing the data from each dataset separately, we observed that the instance hardness approach performed slightly better than traditional classifiers. This indicates that there is potential for further advancement in this methodology. We need to conduct a more in-depth analysis to understand why it only worked for the AssetNegotiation dataset and to investigate the reasons behind the slight increase in accuracy.

REFERENCES

[1] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, "Models and issues in data stream systems," in *Proceedings of the Twenty-First ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, ser. PODS '02. New York, NY, USA: Association for Computing Machinery, 2002, p. 1–16. [Online]. Available: https://doi.org/10.1145/543613.543615

[2] F. Bayram, B. S. Ahmed, and A. Kassler, "From concept drift to model degradation: An overview on performance-aware drift detectors," *Knowledge-Based Systems*, vol. 245, p. 108632, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705122002854

[3] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, "Learning with drift detection," in *Advances in Artificial Intelligence – SBIA 2004*, A. L. C. Bazzan and S. Labidi, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 286–295.

[4] A. Bifet and R. Gavaldà, "Learning from time-changing data with adaptive windowing," vol. 7, 04 2007.

[5] E. Pekalska and R. P. W. Duin, *The Dissimilarity Representation for Pattern Recognition: Foundations And Applications (Machine Perception and Artificial Intelligence)*. USA: World Scientific Publishing Co., Inc., 2005.

[6] J. a. Gama, I. Žliobaitundefined, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Comput. Surv.*, vol. 46, no. 4, mar 2014. [Online]. Available: https://doi.org/10.1145/2523813

[7] M. Salganicoff, "Tolerating concept and sampling shift in lazy learning using prediction error context switching," *Artificial Intelligence Review*, vol. 11, no. 1, pp. 133–155, Feb 1997. [Online]. Available: https://doi.org/10.1023/A:1006515405170

[8] J. Gao, W. Fan, J. Han, and P. S. Yu, *A General Framework for Mining Concept-Drifting Data Streams with Skewed Distributions*, pp. 3–14. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/1.9781611972771.1

[9] S. J. Delany, P. Cunningham, A. Tsymbal, and L. Coyle, "A case-based technique for tracking concept drift in spam filtering," *Knowledge-Based Systems*, vol. 18, no. 4, pp. 187–195, 2005, aI-2004, Cambridge, England, 13th-15th December 2004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705105000316

[10] A. Tsymbal, "The problem of concept drift: definitions and related work," 2004. [Online]. Available: https://api.semanticscholar.org/CorpusID:8335940

[11] Y. Costa, D. Bertolini, A. Jr, G. Cavalcanti, and L. Soares de Oliveira, "The dissimilarity approach: a review," *Artificial Intelligence Review*, vol. 53, 04 2020.

[12] F. A. Pinage and E. M. dos Santos, "A dissimilarity-based drift detection method," in *2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*, 2015, pp. 1069–1076.

[13] S. Basterrech, J. Platoš, G. Rubino, and M. Woźniak, "Experimental analysis on dissimilarity metrics and sudden concept drift detection," in *Intelligent Systems Design and Applications*, A. Abraham, S. Pllana, G. Casalino, K. Ma, and A. Bajaj, Eds. Cham: Springer Nature Switzerland, 2023, pp. 190–199.

[14] M. R. Smith, T. Martinez, and C. Giraud-Carrier, "An instance level analysis of data complexity," *Machine Learning*, vol. 95, no. 2, pp. 225–256, May 2014. [Online]. Available: https://doi.org/10.1007/s10994-013-5422-z

[15] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer, "MOA: massive online analysis," *J. Mach. Learn. Res.*, vol. 11, pp. 1601–1604, 2010. [Online]. Available: https://dl.acm.org/doi/10.5555/1756006.1859903

[16] R. Agrawal, T. Imielinski, and A. Swami, "Database mining: a performance perspective," *IEEE Transactions on Knowledge and Data Engineering*, vol. 5, no. 6, pp. 914–925, 1993.

[17] F. Enembreck, B. C. Ávila, E. E. Scalabrin, and J.-P. Barthès, "Learning drifting negotiations," *Appl. Artif. Intell.*, vol. 21, no. 9, p. 861–881, oct 2007. [Online]. Available: https://doi.org/10.1080/08839510701526954

[18] W. N. Street and Y. Kim, "A streaming ensemble algorithm (sea) for large-scale classification," in *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '01. New York, NY, USA: Association for Computing Machinery, 2001, p. 377–382. [Online]. Available: https://doi.org/10.1145/502512.502568