QuaDapt: Drift-Resilient Quantification via Parameters Adaptation

João Pedro Ortega[1], Luiz Fernando Luth Junior[1], Willian Zalewski[1,2], and André Maletzke[1]

[1] Western Parana State University, Foz do Iguaçu, Brazil
Graduate Program on Electrical Engineering and Computing
{joao.ortega1,luiz.junior90,andre.maletzke}@unioeste.br
[2] Federal University for Latin American Integration, Foz do Iguaçu, Brazil
willian.zalewski@unila.edu.br

Abstract. Quantification is a supervised learning task focused on estimating the prevalence of classes in unlabeled test sets. Although existing quantification methods are effective under prior probability shift, they often fail under more general distribution shift scenarios. In this paper, we introduce `QuaDapt`, a generic and flexible framework that makes classifier-based quantifiers more resilient to distribution shifts. Rather than explicitly modeling specific types of concept drift, `QuaDapt` focuses on adapting the quantifier parameters based on the classifier score distribution observed in the test set. `QuaDapt` updates the quantifier's parameters without requiring labeled test data or retraining. We evaluate `QuaDapt` using artificially generated score distributions to simulate controlled variations in score complexity, as well as on real-world datasets where changes were induced to affect classifier outputs. In artificial experiments, standard quantifiers performed poorly, even as test score distributions became easier. In contrast, `QuaDapt` effectively reduced the quantification error in these scenarios. On 12 real-world datasets with concept drifts, `QuaDapt` consistently enhanced all quantifiers' performance, demonstrating its practical effectiveness and robustness.

Keywords: Dataset shift · Concept drift · Quantification.

1 Introduction

Real-world machine learning systems often face problems in non-stationary environments, where the data observed during deployment differs from the training data [9]. Machine learning methods rely on the assumption that training and test data are drawn from the same underlying distribution. This encompasses assumptions regarding the stability of class priors, class conditional distributions, and marginal distributions. However, these assumptions are often violated in practice, which can significantly degrade the performance and reliability of machine learning models when deployed in real-world environments.

Over the years, the machine learning community has extensively studied distribution shifts across various problem settings [23,26]. In particular, significant

progress has been achieved in the context of classification, encompassing problem formalization as well as detection and adaptation methods for distribution shifts [12,27]. However, in less popular or newly formulated tasks, the understanding of how distribution shifts hinder the models remains poorly explored. One such task is quantification, a supervised learning task formalized by Forman (2005) [10] that aims to estimate the prevalence of each class in an unlabeled test set, rather than assigning labels to individual instances. A straightforward method involves classifying each instance and then counting the number of instances predicted for each class. This method, named Classify and Count (CC), suffers from the systemic error introduced when the class distribution drifts [5].

From Forman's seminal paper [10], a new machine learning community emerged, proposing a wide range of methods to tackle this task. Most of the existing approaches rely on a classifier in a preliminary step to produce either hard predictions or probabilistic scores on the test set. Then, different strategies are employed to infer the class distribution, including simple adjustments based on error rates, probabilistic modeling, and distribution matching [24,11,1,16,22,17].

Although there are contributions to make machine learning tasks more resilient against distribution shift scenarios, a gap remains in the literature regarding strategies that enhance the resilience of quantification under shift scenarios. The most recent contributions on this topic are the works of González et al. (2024) [15] and Maletzke et al. (2021) [20], which analyzed quantification methods under distribution shifts and proposed the first quantification method resilient to distribution shifts, respectively.

In this paper, we propose a generic framework that transforms any classifier-based quantifier into a drift-resilient version by updating its internal parameters using synthetic scores aligned with the estimated test set scores. Similarly to Maletzke et al. (2021), we do not assume any particular form of concept drift or shift. Instead, our framework updates the training-derived posterior probabilities by aligning them with the posterior probabilities observed at deployment through a mixture-model matching process. Based on this alignment, the framework re-estimates the quantifier parameters that depend on these probabilities, such as *tpr*, *fpr*, and the posterior probabilities themselves. Thereby, it adapts the quantifier to the current test conditions without requiring labeled data or retraining.

We evaluate the effectiveness of the proposal in both artificial and real-world scenarios. In the first, we systematically varied the complexity of classifier score distributions using synthetic data. This enabled us to isolate the effect of distribution shifts on quantifier performance. The results reveal a counterintuitive behavior of the current quantifiers, which performed poorly even when the test distributions became easier. In contrast, the `QuaDapt` quantifiers version demonstrated significantly improved robustness, effectively reducing quantification errors. In the second scenario, we show that the proposal consistently enhances the performance of all tested quantifiers. Notably, the `QuaDapt`-adapted quantifiers outperformed their standard counterparts in ten of the 12 datasets, including five datasets where they achieved improvements across all evaluated quantifiers.

The structure of this paper is as follows: Section 2 we present the paper background, including differences between classification and quantification tasks, types of distribution shifts, and a method to tackle quantification in shift scenarios. Section 3 presents the framework proposed for converting classifier-based quantifiers into resilient quantifiers in shift environments. Section 4 depicts the experimental setup and Section 5 presents the empirical results and discussion. Finally, Section 6 concludes this work and presents directions for future work.

2 Background

Quantification is a supervised learning task that aims to estimate the class distribution in an unlabeled test set. This task is closely related to the well-known classification task. For instance, both tasks operate over the same feature representation and assume a nominal output variable that defines the class label. Moreover, many quantification methods depend on a classifier in a preliminary step to produce either hard predictions, probabilistic scores, or error rates, which serve as the basis for estimating the class distribution in the test set [13].

Hence, defining the quantification task requires the classification definition. Let h be a classifier induced from a train set $D = \{(x_1, y_1), ...(x_n, y_n)\}$, where $x_i \in \mathcal{X}$ is a m-dimensional vector in the feature space $\mathcal{X}$ with m attributes and $y_i \in Y = \{c_1, ..., c_l\}$ the respective class label. Therefore, a classifier is defined as a model h induced from D such that [20]:

$$h : \mathcal{X} \rightarrow \{c_1, ...c_l\}$$

In binary tasks, the output space Y comprises two classes, the *positive class* ($c_1 = \oplus$) and the *negative class* ($c_2 = \ominus$). A classifier h aims to assign a class label to unseen instances by thresholding the values provided by a scoring function $h_s : \mathcal{X} \rightarrow \mathbb{R}$ that predicts a numerical value correlated to the posterior probability $P(y_i = \oplus \mid x_i)$. This value indicates the likelihood that x_i belongs to the positive class [10,20].

On the other hand, quantification aims to estimate the prevalence of classes within a given test sample $S \in \mathbb{S}$, rather than assign labels to individual instances as in classification. Accordingly, a quantifier is defined as follows [20]:

$$q : \mathbb{S}^{\mathcal{X}} \rightarrow [0, 1]^l$$

where $\mathbb{S}^{\mathcal{X}}$ represents all the possible samples from $\mathcal{X}$. In a binary scenario, a quantifier estimates a vector $\hat{\mathbf{p}} = \{\hat{p}_{\oplus}, \hat{p}_{\ominus}\}$, where $\hat{p}_{\oplus}$ represents the posterior probability estimation for the positive class, such that $\hat{p}_{\oplus} + \hat{p}_{\ominus} = 1$ [20].

Classify and Count (CC) is one of the simplest quantification methods, estimating class prevalence by applying a classifier to the test set and counting the number of instances predicted for each class. However, despite its simplicity, CC suffers from flagrant shortcomings. The most notable is the systematic error introduced when the class distribution in the test set differs from that of the training data [14]. While CC can yield accurate estimates in the rare case

where the false positive rate (*fpr*) equals the false negative rate (*fnr*), it generally performs poorly under prior shift, which is precisely the scenario where quantification is most relevant. In stationary settings, one could simply use the class proportions from the training set, making quantification unnecessary. Thus, CC's sensitivity to distribution shift severely limits its applicability in real-world quantification tasks.

Although quantification is an emergent task within the machine learning field, several methods have been proposed to overcome the known deficiencies of Classify and Count. In recent years, the community has introduced a variety of strategies aimed at improving quantification accuracy under distribution shift. A well-established taxonomy, proposed by González et al. (2017) [14], organizes these methods into three main categories based on how they leverage classifier outputs and adjust for distributional changes: (*i*) classify, count & correct, (*ii*) adaptation of classification algorithms, and (*iii*) distribution matching.

In this paper, we focus on quantification methods that rely on a classifier in previous steps, aiming to analyze their behavior under distribution shift scenarios and propose a framework to enhance their robustness. Therefore, we restrict our focus on methods from the group (*i*) and (*iii*) that are summarized in Table 1, along with their respective descriptions and references.

Table 1: Quantification taxonomy

Group	Quantifier	Brief Description	Reference
I	CC	Classify & Count	[10,11]
	ACC	Adjusted Classify & Count	
	X	Classifier with decision threshold adjusted such that fnr[1] $= fpr$[2]	
	MAX	Classifier with decision threshold chosen such that tpr[3] $= fpr$ is maximum	
	T50	Classifier with decision threshold adjusted such that $tpr = 50\%$	
	MS	Median of Classify & Count computed along all decision threshold values	
III	HDy	Mixture model with Hellinger distance	[16]
	DyS	Framework for Mixture models	[22]
	SMM	Sample Mean Matching	[17]

Although widely used and recognized in the quantification literature for their effectiveness in estimating class prevalence, most existing methods are explicitly designed to handle variations in the prior class distribution $P(Y)$. However, despite their widespread adoption in many studies [1,21,8,17,25,6,18], these methods often overlook other important forms of distribution shift, which are common in real-world applications and can substantially impair quantifier performance.

This oversight represents a significant limitation, as real-world data is seldom stationary and often undergoes complex, evolving shifts that can compromise the reliability of classifier-based quantifiers. González et al. (2024) [15] recently

[1] False Negative Rate (*fnr*): proportion of positives incorrectly predicted as negatives.
[2] False Positive Rate (*fpr*): proportion of negatives incorrectly predicted as positives.
[3] True Positive Rate (*tpr*): proportion of positives correctly predicted as positives.

highlighted these issues, emphasizing the need to better understand and address the vulnerabilities of quantification methods in the face of a broader range of distribution shifts. In the following section, we will present various types of distribution shifts that can severely impact the performance of current quantifiers.

2.1 Distribution Shifts

Supervised learning methods typically assume that training and test data are drawn from the same distribution. However, this assumption rarely holds in real-world scenarios, where changes in the underlying data are common. In the context of quantification, the primary focus has been on handling prior probability shift, where the class distribution $P(Y)$ changes between training and test sets. While many quantifiers have been designed under this assumption, they often neglect other types of shifts, such as changes in the feature distribution $P(X)$ or the class-conditional distribution $P(X|Y)$. These additional forms of shifts can severely degrade quantification accuracy, especially for methods that rely on classifier-derived information. As highlighted by González et al. (2024) [15], the performance of many state-of-the-art quantifiers deteriorates substantially under realistic drift scenarios, revealing a critical need for more resilient approaches.

According to González et al. (2024) [11], the following changes can affect quantification methods:

Prior Probability Shift: refers to changes in the class distribution between training and test data. Formally, this condition is characterized by $P^{\text{tr}}(Y) \neq P^{\text{ts}}(Y)$, where $P^{\text{tr}}(Y)$ and $P^{\text{ts}}(Y)$ denote the class distributions in the training and test sets, respectively.

Covariate Shift: refers to changes in the marginal distribution of features changes between training and test sets, formally $P^{\text{tr}}(X) \neq P^{\text{ts}}(X)$, while the posterior class probabilities remain stable, i.e., $P^{\text{tr}}(Y|X) = P^{\text{ts}}(Y|X)$. This implies that the predictive relationship between features and labels is preserved, even though the distribution of features itself varies.

Concept Drift: refers to changes in the relationship between features and classes, specifically $P(Y|X)$, which can also affect $P(X|Y)$. This drift can be particularly challenging to detect and manage, as it implies that the underlying patterns that link features to classes have evolved over time [15].

These data shifts can significantly impact the outcomes of quantification models, due to their effect on the scores assigned by a *scorer*. Maletzke et al. (2021) [20] analyze the impact of distributional changes on quantification by examining how such shifts affect the complexity of score distributions produced by binary classifiers. In their seminal work, they demonstrate that even when changes simplify the underlying decision boundary, effectively making the classification task easier, most quantification methods (with the exception of CC) still suffer from significant performance degradation. This counterintuitive behavior reveals a critical vulnerability and highlights the lack of robustness in existing quantifiers when faced with shifts that alter the complexity of score distribution.

In [20], the authors proposed DySyn, a quantification method resilient to distribution shifts based on the quantifier DyS [22] and the method Model for Score Simulation (MoSS). In this paper, we extend the work of Maletzke et al. (2021) [20] by proposing a general framework that transforms any classifier-based quantifier into a distribution shift–resilient quantifier. MoSS and DySyn are described in the next sections.

2.2 MoSS

Classification scores can be evaluated to detect mismatches between training and test data. In binary classification, if the classifier accurately models the concept of each class, then the scores generated by the classifier for each class will be far apart from each other. Maletzke et al. (2021) [20] noted in their paper that, in some circumstances, when a distribution shift occurs during the deployment phase, the predicted score distributions change, impacting the accuracy of quantifiers. They proposed modeling a new training score distribution based on the predicted score distribution from the deployment phase and then reapplying the quantifier. To model new training scores, Maletzke et al. (2021) [20] proposed the Model for Score Simulation (MoSS).

The MoSS model produces two synthetic distributions, simulating the scores of the positive and negative class labels, parameterizing the overlap between the scores of each class. Thus, beyond generating artificial scores, parameterization enables MoSS to produce scores that emulate different scenarios of complexity between the positive and negative classes.

MoSS relies on three parameters: the *merging* factor $\mathfrak{m}$, which ranges from 0 to 1, controlling the overlap degree between the positive and negative scores (0 representing the easiest scenario with well-separated scores, while 1 indicates the most challenging case with highly overlapping scores), the number of observations n, and α that defines the proportion of the positive class [20]. The MoSS model is presented in the following equation:

$$\text{MoSS}(n, \alpha, \mathfrak{m}) = \text{syn}(\oplus, \lfloor \alpha n \rfloor, \mathfrak{m}) \cup \text{syn}(\ominus, \lfloor (1-\alpha) n \rfloor, \mathfrak{m})$$

where,

$$\begin{array}{ll} \text{syn}(\oplus, n, \mathfrak{m}) = \bigcup_{i=1}^{n} \{X_i^{\mathfrak{m}}\}, & X_i \sim U(0,1) \\ \text{syn}(\ominus, n, \mathfrak{m}) = \bigcup_{i=1}^{n} \{1 - X_i^{\mathfrak{m}}\}, & X_i \sim U(0,1) \end{array}$$

A synthetic score in MoSS can be defined as a non-linear map of a uniformly distributed random variable ranging in $[0, 1]$, enabling flexible generation of different degrees of overlap between positive and negative scores. This synthetic generation process plays a central role in the DySyn method, which we present in the following section.

2.3 DySyn

In [20], the authors explored quantification to deal with distribution shifts by modifying methods based on the test set. The authors proposed the quantifi-

cation algorithm named as Distribution y-Similarity (DySyn) using Synthetic Scores. DySyn combines the DyS framework and the MoSS model.

DyS aims to find the optimal mixture of positive and negative scores ($S^{\text{tr}}_{\oplus}$ and $S^{\text{tr}}_{\ominus}$, respectively), estimated from the training set, by searching for the α (proportion of the positive class) value that minimizes the distance function DS between the resulting mixed distribution and the test set distribution $S^{\odot}$. The DyS is defined as follows:

$$\text{D}y\text{S}(S^{\text{tr}}_{\oplus}, S^{\text{tr}}_{\ominus}, S^{\odot}) = \underset{0\leq\alpha\leq 1}{\textbf{arg min}} \left\{ \text{DS}\left(\alpha R[S^{\text{tr}}_{\oplus}] + (1-\alpha)R[S^{\text{tr}}_{\ominus}], R[S^{\odot}]\right)\right\}$$

where $S^{\odot}$, R, and DS are the scores of the test set, a representation function, and a distance function to operate over R representation, respectively.

In summary, DySyn runs an additional search across multiple values of the MoSS merging factor($\mathfrak{m}$), selecting the one that generates the mixed score distributions most similar to the test scores according to a distance function. This adaptive mechanism enables DySyn to align with the underlying score distribution of the test data, making it more robust against distribution shifts. DySyn is formally defined as follows:

$$\text{D}y\text{Syn}(S^{\odot}) = \underset{\substack{0\leq\alpha\leq 1\\ \text{s.t. } 0\leq\mathfrak{m}\leq 1}}{\textbf{arg min}} \left\{ \text{DS}\left(R[\text{MoSS}(n,\alpha,\mathfrak{m})], R[S^{\odot}]\right)\right\}$$

2.4 Related Works

Quantification methods are designed to address *prior probability shift*, which refers to changes in the class distribution $P(Y)$ between training and test data. However, *other types of distribution shift*, such as changes in the feature distribution $P(X)$ or in the class-conditional distribution $P(X|Y)$, can also affect machine learning models, including quantifiers. While these types of shifts have been extensively studied in the *classification* literature, they are overlooked in the context of quantification.

This gap remains poorly explored, with only a few recent efforts addressing the effects of broader distribution shifts. Maletzke et al. (2021) [20] discussed in their research the flaws of classic quantification methods. To address the limitations of standard quantifiers, Maletzke et al. (2021) [20] proposed an innovative method referred to as DySyn, which modifies DyS utilizing MoSS to identify an optimal fit for the score distribution based on a new dataset from a test set, rather than relying on training scores. DySyn was evaluated across 12 real-world datasets, outperforming 15 state-of-the-art quantifiers.

González et al. (2024) [15] assessed the performance of different quantification methods under three types of drift: (1) prior probability shift, (2) covariate shift, and (3) concept shift. The authors demonstrate that while literature quantifiers are resilient to prior probability shifts, they perform poorly under other types of drift, and this issue remains underexplored.

3 Proposal

In this paper, we extend the D*y*Syn method, showing that it is an instance of a more general and non-formalized framework, enabling any classifier-based quantifier to adapt its behavior without requiring access to labeled test data or retraining, thus ensuring robustness in real-world non-stationary environments.

To formalize the framework, consider q a binary quantifier that predicts the class prevalence $\hat{p}$ in the test set as follows:

$$\hat{p} = q(h_S(S^{\odot}), \Theta)$$

where h_S, $S^{\odot}$, and Θ represent the scorer, the unlabeled test set, and the quantifier parameters, respectively. Quantifier parameters Θ vary according to the quantification method, ranging from classifier performance metrics to class-conditional scores in distribution matching methods. For the sake of clarity, we categorize the quantifier parameters based on the taxonomy of quantification, as follows:

- For **classify, count & correct** methods (e.g., ACC and MS), include classifier performance metrics such as true positive rate (*tpr*) and false positive rate (*fpr*), which are typically derived from the training set using a decision threshold τ.
- For **distribution matching** methods (e.g., D*y*S and HDy), Θ consists primarily of class-conditional scores. Additionally, score representations (e.g., histograms or kernel functions) and their respective hyperparameters (e.g., the number of bins and kernel type), along with the similarity function, are also parameters in this category.

In both groups, the **class-conditional scores** represent a fundamental parameter, i.e., their degree of separation determines the difficulty of the quantification problem. Well-separated class-conditional scores correspond to an easier scenario, which typically translates into higher *tpr* and lower *fpr* for correction-based methods, as well as into clearer and more discriminative score representations for distribution-matching methods.

As noted by Maletzke et al. (2020) [21] and Maletzke et al. (2018) [19], concept drift is often manifested as changes in the complexity of class-conditional scores, since shifts in the data distribution directly affect how the classifier separates positive and negative instances. For this reason, in this paper, we consider the class-conditional scores as the primary parameters for both groups of quantifiers, while keeping the remaining parameters, such as threshold, similarity functions, or representation formats, fixed.

For a given training set $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, the positive and negative class-conditional scores can be obtained through k-fold cross-validation. Therefore, a generic function to obtain the remaining quantifier parameters can be defined as follows:

$$\Theta = \Psi(S^{\text{tr}}_{\oplus},\ S^{\text{tr}}_{\ominus},\ \tau)$$

where τ represents the classification threshold used on the class-conditional scores for calculating performance measures, true positive rate (*tpr*), and the false positive rate (*fpr*). For example, given the training score distributions $S^{\text{tr}}_{\oplus}$ and $S^{\text{tr}}_{\ominus}$, different choices of τ (e.g., decision thresholds) will produce distinct values for *tpr* and *fpr*, impacting on the quantifiers accuracy.

In contrast to relying on fixed training-derived parameters, Maletzke et al. (2021) [20] noted that modifying the training scores of the D*y*S method based on the test set leads to more resilient results under drift scenarios. Based on this assumption, we propose **Drift-Resilient Quantification via Parameters Adaptation** (`QuaDapt`), a generic framework for quantifying in dynamic environments. Our framework, `QuaDapt`, concentrates on updating the **class-conditional scores** parameter, which represents the core parameter across both quantification methods groups. By realigning these scores with the characteristics of the test set, `QuaDapt` effectively influences both **classify, count & correct** methods, where performance measures such as *tpr* and *fpr* depend on score distributions, and **distribution matching** methods, where prevalence is inferred by comparing score distributions directly. Therefore, adapting class-conditional scores serves as a general mechanism that enhances the robustness of quantifiers across both categories under drift scenarios.

Figure 1 provides a conceptual overview of how the `QuaDapt` framework operates. Starting from a labeled training set, a scorer h_S is fitted to produce posterior scores. During deployment, when an unlabeled test set arrives, the scorer outputs the test scores $S^{\odot}$. Instead of relying solely on the training-derived scores, `QuaDapt` generates multiple sets of synthetic scores using MoSS, varying the mixing factor $\mathtt{m}$ to represent different levels of class separability. The framework then searches for the synthetic distribution that minimizes the dissimilarity (DS) with the observed test scores $S^{\odot}$. The selected synthetic scores $\hat{S}_{\oplus}$ and $\hat{S}_{\ominus}$ are finally used to re-estimate the quantifier parameters $\hat{\Theta}$, which include correction factors such as *tpr*, *fpr*, and D*y*S's histograms. By realigning parameters in this way, `QuaDapt` adapts the quantifier to the actual test conditions, ensuring robustness under distribution shifts.

`QuaDapt` is formally defined as follows:

$$\hat{p} = \mathcal{Q}_{\texttt{QuaDapt}}\left(q[h_S(S^{\odot}), \hat{\Theta}]\right)$$

where q is a quantifier that requires a classifier or scorer as a previous step, and Θ represents the q parameters updated by the same Ψ procedure, as follows:

$$\hat{\Theta} = \Psi(\hat{S}_{\oplus},\ \hat{S}_{\ominus},\ \tau)$$

where:

– $\hat{S}_{\oplus}, \hat{S}_{\ominus}$: synthetic scores for the positive and negative classes, respectively, generated using MoSS with a mixing factor $\mathtt{m}$. The mixing factor is selected minimizing the dissimilarity (DS) between the combined synthetic distribu-

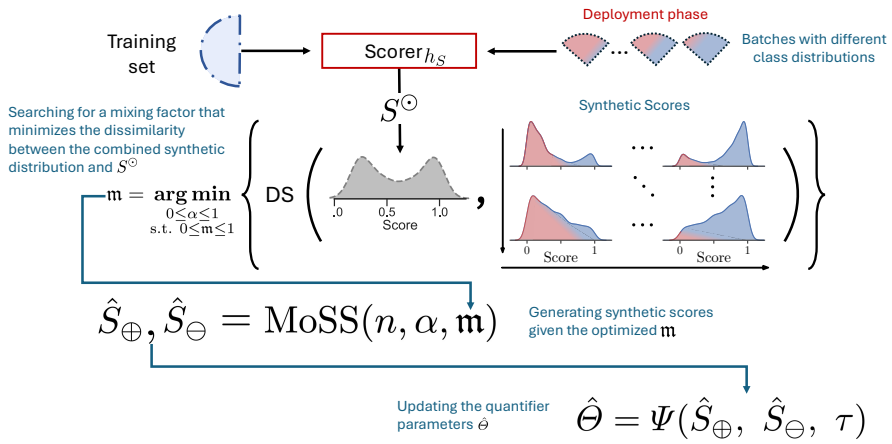

Fig. 1: Schematic representation of `QuaDapt`. Parts of this figure are adapted from [20].

tion and the distribution of test scores $S^{\odot}$, defined as follows:

$$\mathfrak{m} = \underset{\substack{0\leq\alpha\leq1 \\ \text{s.t. } 0\leq\mathfrak{m}\leq1}}{\arg\min} \{\text{DS}\left(R[\text{MoSS}(n, \alpha, \mathfrak{m})], R[S^{\odot}]\right)\}$$

where R and n are the representation function (e.g., histogram or Cumulative Distribution Function) and number of synthetic scores generated by MoSS;

– τ: same decision threshold used in training-based estimation, required to compute classifier metrics such as *tpr* and *fpr*.

4 Experiments

To evaluate our framework, we adopt the experimental setup introduced by Maletzke et al. (2021) [20]. Our evaluation is organized into two experiments, each designed to analyze the behavior of quantifiers under different conditions of distribution shift, reflected in the complexity of classifier scores.

In the **first experiment**, we leverage artificial score generation, using the MoSS model, to produce test scores with varying levels of complexity between training and test phases. A key advantage of using MoSS is that it allows us to abstract away from the specific source of data drift (e.g., covariate shift or concept drift). This setup enables a controlled evaluation of how well quantifiers can adapt to changes in score complexity, even in cases where the test-time classification task becomes easier. Thus, when the decision boundary becomes simpler, the expected result is improved quantifier performance, as observed in the classification task.

In the **second experiment**, we extend the analysis to a collection of real-world datasets. We explicitly enforce concept drift by modifying the test data so that the classification task becomes either simpler or more challenging relative to

the training data. The goal is to assess how the quantification methods perform in realistic scenarios.

In both experiments, we evaluated the quantifiers listed in Table 1, ensuring consistency across experimental conditions. The `QuaDapt` framework adjusts classifier-based quantifiers by aligning artificial positive and negative score distributions with those observed in the test set, using a MoSS merging factor $\mathtt{m}$ varied from 10% to 90% in 20% increments to generate diverse training score distributions. The performance of each method was assessed using the Mean Absolute Error (MAE), which quantifies the average absolute difference between the predicted and true prevalence of the positive class. Additionally, for the DyS method, the Topsøe distance was used as the divergence measure to guide its internal optimization process.

4.1 Experiment 1 - Artificial Scores

Our first experiment was designed to evaluate the existing quantification methods and the proposed framework under drift scenarios, without requiring manipulation of real datasets to induce distribution shifts.

Using MoSS, we generate synthetic training and test scores with varying levels of complexity, adjusting the merging factor $\mathtt{m}$. Thus, we systematically control the separation between positive and negative score distributions, enabling the simulation of both easier and harder decision boundaries. This setup allows us to evaluate quantification performance across a range of drift scenarios where the score distributions differ between training and testing at controlled intensities, without being tied to a specific type of real-world shift.

In the training phase, we generated a total of 2,000 synthetic scores, 1,000 for the positive class ($\oplus$) and 1,000 for the negative class ($\ominus$), varying the MoSS merging factor for training ($\mathtt{m}_{\mathrm{tr}}$) from 5% to 95% in increments of 5%. For each training configuration defined by $\mathtt{m}_{\mathrm{tr}}$, we then evaluated multiple test scenarios by varying the following parameters in the testing phase:

- **Test size**: fixed at 100 instances, i.e., 100 score values per test set;
- **Class proportion** : from 0% to 100% in increments of 1%;
- **MoSS merging factor for test** ($\mathtt{m}_{\mathrm{ts}}$): varied from 5% to 95% in increments of 5%, simulating of test score distributions with different complexities;
- **Replicates**: each configuration was randomly repeated 10 times.

4.2 Experiment 2 - Real-world datasets

Our second experiment was designed to assess the performance of existing quantification methods and the proposed `QuaDapt` framework on real-world datasets. We adopt the experimental setup originally described in [20] to evaluate the standard quantifiers and quantifiers integrated into the `QuaDapt` framework.

Table 2 presents a brief description of the datasets. The datasets were obtained from UCI [4] and OpenML [28] repositories[3]. As performed by Maletzke

[3] Specific citations are requested for Avila [3] and Walking [2]. Jock A. Blackard and Colorado State University preserve copyright over Covertype.

et al. (2021) [20], we use multi-class datasets that are converted into binary problems by grouping the classes into positive and negative classes. The selection of class labels for the positive and negative classes was conducted strategically to create a subset of positive classes and two distinct subsets of negative classes, referred to as **easier** and **harder**. This setup was designed to simulate scenarios with varying complexity, thereby enabling the evaluation of quantification methods under different degrees of concept drift.

Table 2: Datasets description [20].

Dataset	Classes	Positive	Negative Class					
			Easier Subclass			Harder Subclass		
			Classes Name	Overlap (%)	F1 Score	Classes Name	Overlap (%)	F1 Score
Avila	12	[A]	[I]	0.43	0.99	[E,F,G,H,X]	6.83	0.86
Chess game	18	[fourteen]	[zero,...,nine]	3.05	0.98	[thirteen]	21.33	0.82
Covertype	7	[Ponderosa_Pine]	[Krummholz]	5.06	0.94	[Doug.,Aspen,Cott.]	23.69	0.81
Dermatology	6	[6]	[3]	0.00	1.00	[1,4,5]	0.30	0.97
HAR	6	[5]	[6]	0.00	1.00	[4]	7.45	0.93
Land-use	8	[Corn]	[Hay]	0.16	1.00	[Soybeans]	20.76	0.74
MFeat	10	[3,5]	[2,9,10]	2.54	0.99	[4,7,8]	19.31	0.81
Mosquitoes	3	[*Ae.aegypti* ♀]	[*An.aquasalis* ♂]	2.31	0.98	[*Cx.quinquefasciatus* ♀]	27.44	0.76
Nursery	5	[priority]	[not_recom]	0.00	1.00	[very_recom,spec_prior]	9.15	0.92
Phishing URL	5	[Defacement]	[benign]	1.38	0.99	[malware,phishing]	4.35	0.98
Satimage	6	[4]	[2]	0.46	0.99	[3,5,7]	19.74	0.80
Walking	22	[6]	[10]	1.22	0.99	[9,12]	43.39	0.79

To build the positive and negative classes (**easier** and **harder**), each dataset is split into training and test sets. A Random Forest classifier is then fit on the training set, and the predicted scores are obtained for the test set. We then computed the overlapped area between the positive and negative score distributions. A smaller overlap area indicates a simpler scenario, while a larger overlap reflects greater ambiguity between classes and thus a more complex setup. This procedure was systematically repeated to identify the easier and harder negative class groupings for each dataset. In addition, we computed the F1-Score to illustrate the difference between easier and harder scenarios, providing a complementary view of difficulty beyond overlap measurements.

Columns three, four, and seven in Table 2 represent the subclasses that compose the positive and negative classes, including the **easier** and **harder** negative subclasses for each dataset. Columns five, six, eight, and nine illustrate the intersection of the **easier** and **harder** negative subclasses with the positive class, respectively.

Therefore, after defining positive and negative classes, along with **easier** and **harder** subclasses, we sample each dataset according to the same distribution for every class. Then we split each dataset into two halves: training and test sets. We perform stratified sampling to preserve the distribution of the positive and negative subclasses. In the training portion, we apply 10-fold cross-validation to generate the score distributions required by the mixture models and to estimate the *tpr* and *fpr* rates employed by the quantifiers from the Classify, Count, and Correct group. We generate all classifier scores using Random Forests with 200 trees. In addition, we train a single Random Forest model, using the entire training set, to produce the test scores used in the evaluation phase.

To simulate realistic drift scenarios in our real-world experiment, we adhered to the Artificial Prevalence Protocol (APP) [7,24] by extracting multiple samples from the test set and systematically varying the prevalence of both positive and negative subclasses. We explored different combinations of **easier** and **harder** negative subclasses to create test distributions with varying levels of complexity. The configuration space for this experimental setup is detailed below:

- **Positive Class Proportion**: varied from 0% to 100% in increments of 1%;
- **Harder Negative Subclass Proportion**: from 0% to 100% in increments of 25%;
- **Test Set Size**: fixed at 100 instances per test sample;
- **Replicates**: each configuration was randomly repeated 10 times.

The quantifiers were assessed by conducting a statistical comparison between the original methods and their `QuaDapt` version. We utilized a paired two-tailed t-test to determine whether the observed differences in performance, measured by MAE, were statistically significant for each dataset.

5 Results

We open this section presenting the results of our **first experiment**, designed to assess quantifiers performance under controlled variations in score complexity. Although both $\mathfrak{m}_{\text{tr}}$ and $\mathfrak{m}_{\text{ts}}$ were independently varied across a wide range, we present plots of MAE as a function of $\mathfrak{m}_{\text{ts}}$ for only four selected fixed values of $\mathfrak{m}_{\text{tr}}$. This allows us to illustrate representative patterns while maintaining clarity in visual analysis. These results are presented in Figure 2.

Solid red lines indicate the performance of the standard quantifiers, while dashed green lines illustrate the results of their corresponding `QuaDapt`-adapted versions. The solid blue line represents the baseline CC method. The vertical dashed lines mark the point where $\mathfrak{m}_{\text{tr}} = \mathfrak{m}_{\text{ts}}$.

To the right of the vertical line ($\mathfrak{m}_{\text{ts}} > \mathfrak{m}_{\text{tr}}$), the problem becomes increasingly difficult, as the overlap between class score distributions grows. In these scenarios, MAE tends to rise for both standard and adapted methods, which is expected. However, to the left of the vertical line ($\mathfrak{m}_{\text{ts}} < \mathfrak{m}_{\text{tr}}$), the test score distributions become more separable, making the classification problem easier. In such cases, *standard quantifiers surprisingly continue to exhibit poor performance*, failing to exploit the increased separability.

This undesirable behavior of standard quantifiers under easier test conditions was also observed by Maletzke et al.(2021) [20]. However, the results in Figure 2 show that our proposal transforms classifier-based quantifiers into more resilient quantifiers when faced with these favorable changes. This represents a significant improvement, as the inability to take advantage of easier scenarios represents a serious limitation in practical applications of quantification.

Our **second experiment** aims to evaluate the effectiveness of our framework under realistic conditions. We adopted the experimental protocol described in [20]. Table 3 reports the MAE for each quantifier across 12 datasets. For each

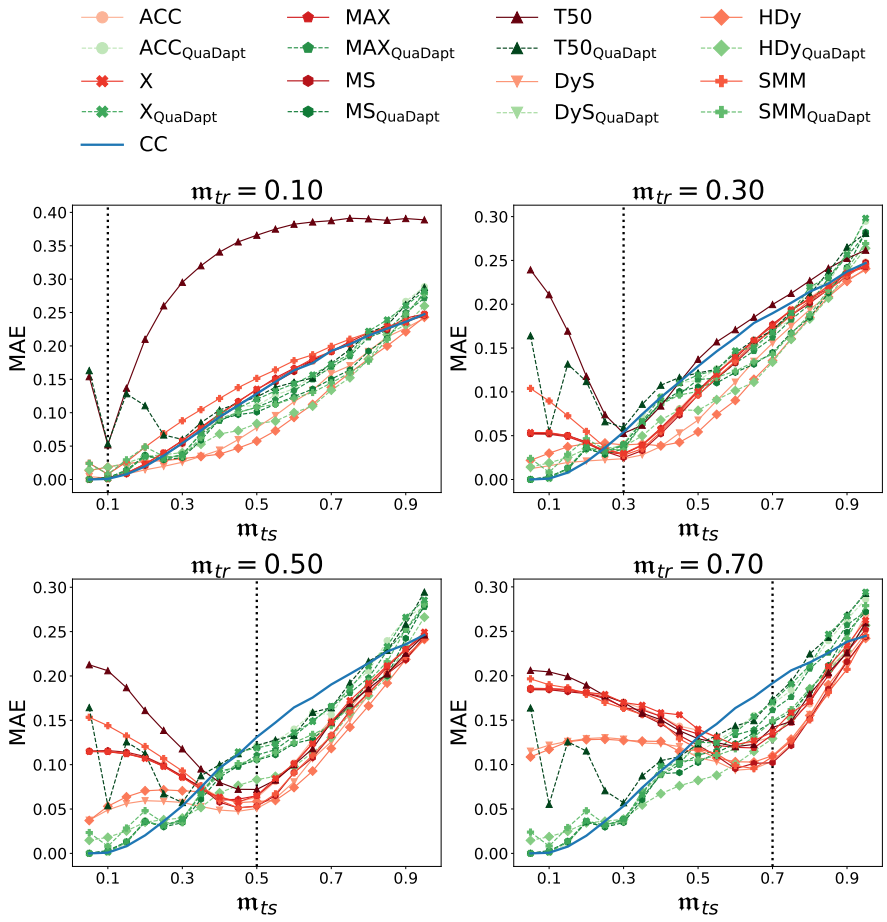

Fig. 2: MAE for fixed values of $\mathfrak{m}_{\text{tr}} \in \{0.10, 0.30, 0.50, 0.70\}$, as $\mathfrak{m}_{\text{ts}}$ varies. Vertical dashed lines indicate the point where $\mathfrak{m}_{\text{tr}} = \mathfrak{m}_{\text{ts}}$.

quantifier, we present results in pairs: the first line corresponds to the original (unadapted) quantifier, and the second line to its `QuaDapt` version. The best MAE values for each dataset are shown in **bold**, with the corresponding standard deviation reported in parentheses. Results that are statistically better ($p < 0.05$) are underlined.

The row `QuaDapt` **win rate** shows, for each dataset, the percentage and absolute number of cases where the `QuaDapt` quantifiers outperformed their standard counterparts in terms of MAE.

Notably, in five out of the twelve datasets, our proposal improved the performance of all evaluated quantifiers (win rate = 100%), demonstrating consistent effectiveness across diverse scenarios. Furthermore, in ten out of twelve datasets, the `QuaDapt` framework enhanced the performance of at least half of the evalu-

Table 3: Mean absolute error of standard quantifiers and their corresponding QuaDapt versions across all datasets.

Quantifiers	Avila	Chess game	Covertype	Dermatology	HAR	Land-use
Baseline (CC)	0.056 (0.039)	0.045 (0.033)	0.077 (0.053)	0.002 (0.005)	0.017 (0.015)	0.061 (0.041)
ACC	0.065 (0.044)	0.052 (0.037)	0.059 (0.045)	0.002 (0.005)	0.026 (0.020)	0.084 (0.059)
$ACC_{QuaDapt}$	**0.046 (0.054)**	**0.031 (0.026)**	**0.057 (0.055)**	0.002 (0.005)	**0.017 (0.015)**	**0.048 (0.037)**
X	0.060 (0.041)	0.046 (0.033)	0.070 (0.054)	0.003 (0.006)	0.020 (0.014)	0.071 (0.046)
$X_{QuaDapt}$	**0.046 (0.054)**	**0.030 (0.027)**	**0.056 (0.054)**	**0.002 (0.005)**	**0.017 (0.015)**	**0.048 (0.037)**
MAX	0.070 (0.061)	0.046 (0.038)	0.071 (0.057)	0.002 (0.005)	0.020 (0.014)	0.085 (0.075)
$MAX_{QuaDapt}$	**0.050 (0.061)**	**0.034 (0.030)**	**0.062 (0.064)**	0.002 (0.005)	**0.017 (0.015)**	**0.052 (0.044)**
T50	0.153 (0.110)	0.127 (0.088)	**0.073 (0.059)**	**0.089 (0.067)**	0.164 (0.102)	0.160 (0.110)
$T50_{QuaDapt}$	**0.094 (0.092)**	**0.088 (0.086)**	0.121 (0.104)	0.211 (0.127)	**0.144 (0.103)**	**0.091 (0.085)**
MS	0.156 (0.118)	0.109 (0.071)	**0.085 (0.063)**	**0.022 (0.019)**	0.119 (0.088)	0.143 (0.097)
$MS_{QuaDapt}$	**0.074 (0.060)**	**0.060 (0.049)**	0.096 (0.070)	0.120 (0.106)	**0.073 (0.044)**	**0.063 (0.052)**
HDy	0.079 (0.055)	0.048 (0.036)	0.072 (0.057)	**0.008 (0.008)**	0.038 (0.030)	0.102 (0.064)
$HDy_{QuaDapt}$	**0.049 (0.047)**	**0.039 (0.031)**	**0.059 (0.050)**	0.050 (0.044)	**0.026 (0.021)**	**0.038 (0.033)**
DyS	0.065 (0.047)	0.035 (0.029)	0.062 (0.050)	**0.002 (0.005)**	0.016 (0.013)	0.065 (0.051)
DyS$_{QuaDapt}$	**0.044 (0.051)**	**0.030 (0.027)**	**0.060 (0.054)**	0.010 (0.010)	**0.013 (0.011)**	**0.038 (0.033)**
SMM	0.102 (0.063)	0.070 (0.044)	**0.079 (0.058)**	**0.006 (0.006)**	0.053 (0.035)	0.114 (0.070)
$SMM_{QuaDapt}$	**0.090 (0.068)**	**0.065 (0.048)**	0.091 (0.064)	0.036 (0.023)	**0.048 (0.025)**	**0.065 (0.053)**
QuaDapt **win rate**	100% (8/8)	100% (8/8)	63% (5/8)	13% (1/8)	100% (8/8)	100% (8/8)
Quantifiers	**Mfeat**	**Mosquitoes**	**Nursery**	**Phishing URL**	**Satimage**	**Walking**
Baseline (CC)	0.052 (0.036)	0.062 (0.043)	0.022 (0.018)	0.018 (0.015)	0.034 (0.023)	0.098 (0.066)
ACC	0.058 (0.039)	0.060 (0.042)	0.027 (0.020)	0.026 (0.018)	0.090 (0.064)	**0.062 (0.049)**
$ACC_{QuaDapt}$	**0.042 (0.032)**	**0.045 (0.033)**	0.027 (0.025)	**0.018 (0.015)**	**0.034 (0.023)**	0.073 (0.056)
X	0.058 (0.038)	0.055 (0.039)	**0.026 (0.018)**	0.019 (0.014)	0.058 (0.036)	**0.070 (0.059)**
$X_{QuaDapt}$	**0.042 (0.032)**	**0.045 (0.033)**	0.027 (0.025)	**0.018 (0.015)**	**0.033 (0.023)**	0.072 (0.056)
MAX	0.048 (0.036)	0.054 (0.042)	**0.025 (0.018)**	0.018 (0.014)	0.051 (0.035)	0.078 (0.067)
$MAX_{QuaDapt}$	**0.044 (0.033)**	**0.048 (0.036)**	0.028 (0.027)	0.018 (0.015)	**0.034 (0.023)**	**0.077 (0.065)**
T50	0.075 (0.056)	0.111 (0.081)	**0.120 (0.082)**	**0.049 (0.042)**	0.167 (0.115)	0.066 (0.055)
$T50_{QuaDapt}$	**0.069 (0.055)**	**0.106 (0.086)**	0.161 (0.136)	0.103 (0.074)	**0.053 (0.046)**	**0.063 (0.058)**
MS	0.064 (0.043)	0.097 (0.065)	0.084 (0.054)	0.036 (0.027)	0.144 (0.100)	0.074 (0.062)
$MS_{QuaDapt}$	0.064 (0.045)	**0.071 (0.055)**	**0.074 (0.058)**	**0.020 (0.017)**	**0.054 (0.038)**	**0.070 (0.052)**
HDy	0.039 (0.031)	0.055 (0.041)	**0.031 (0.024)**	**0.017 (0.014)**	0.064 (0.043)	0.073 (0.062)
$HDy_{QuaDapt}$	**0.035 (0.029)**	**0.042 (0.031)**	0.050 (0.036)	0.025 (0.021)	**0.038 (0.039)**	**0.055 (0.047)**
DyS	0.038 (0.029)	**0.040 (0.030)**	**0.020 (0.018)**	0.014 (0.012)	0.035 (0.025)	**0.066 (0.058)**
DyS$_{QuaDapt}$	**0.037 (0.027)**	0.048 (0.033)	0.027 (0.023)	**0.013 (0.011)**	**0.017 (0.013)**	0.067 (0.053)
SMM	**0.051 (0.035)**	0.070 (0.046)	0.051 (0.032)	0.023 (0.017)	0.090 (0.057)	**0.071 (0.056)**
$SMM_{QuaDapt}$	0.059 (0.038)	**0.067 (0.044)**	**0.049 (0.036)**	**0.019 (0.014)**	**0.049 (0.027)**	0.083 (0.063)
QuaDapt **win rate**	75% (6/8)	88% (7/8)	25% (2/8)	63% (5/8)	100% (8/8)	50% (4/8)

ated quantifiers, underscoring its robustness to provide broad and reliable performance gains across different data distributions and drift conditions.

Among the quantifiers, $X_{QuaDapt}$ achieved the highest win rate, improving its standard version in 83% of the datasets, followed by $ACC_{QuaDapt}$, $MAX_{QuaDapt}$, $MS_{QuaDapt}$, and $HDy_{QuaDapt}$, each with a 75.0% win rate. For $T50_{QuaDapt}$, DyS$_{QuaDapt}$, and $SMM_{QuaDapt}$, the framework improved their performance in 67%. On average, across all datasets and quantifiers, the framework provided an improvement of 73%, highlighting its consistent ability to enhance quantification performance under distribution shifts.

Dermatology and Nursery datasets emerged as special cases in our evaluation. Both datasets present a very high F1-scores for the harder subclasses (0.97 and 0.92), indicating that even their **harder** scenarios remain relatively simple, with well-separated classes and low ambiguity. As a result, standard quantifiers already achieve near-optimal accuracy, leaving limited space for QuaDapt to provide improvements. Nevertheless, the proposal achieved competitive results for some quantifiers, such as $ACC_{QuaDapt}$, $MAX_{QuaDapt}$, and $X_{QuaDapt}$, confirming that even in easy scenarios where performance margins are narrow, QuaDapt can still match standard methods. In summary, the poor win rates observed in these datasets (13% and 25%) reflect the absence of meaningful drift challenges rather than a shortcoming of the framework itself.

Due to the lack of space, Figure 3 presents the results[4] for only five quantifiers for Avila and Chess game datasets, where the contribution of our proposal was most evident. This figure shows the performance of five quantifiers as the proportion of the harder subclass increases. Larger proportions of the harder subclass lead to greater overlap between classes, thereby making the task progressively more difficult. The vertical dashed line at 0.5 indicates the scenario where training and test sets have the equivalent complexity. For both selected datasets, traditional quantification methods tend to exhibit optimal performance under conditions akin to those undergone during training. However, their effectiveness diminishes even in less challenging scenarios, surprisingly. In contrast, the `QuaDapt` methods show greater resilience in different conditions, maintaining more stable performance in both easier and harder settings.

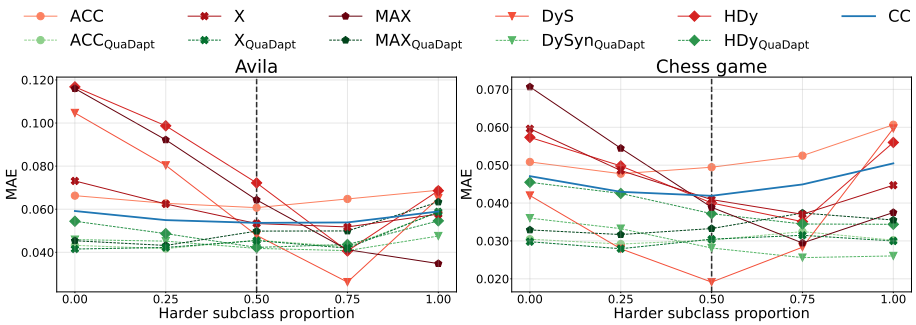

Fig. 3: Quantifiers MAE by harder subclass proportion.

In general, our results on real-world datasets are consistent with the findings from our first experimental setup using artificial scores. Specifically, methods that rely strictly on score distributions and correction rates derived from the training data tend to perform poorly when there is a mismatch between the training and testing distributions, a condition commonly observed under distribution shifts. This degradation occurs even when the test distribution becomes more separable, which makes the task easier.

We highlight a fundamental contrast between classification and quantification under changing data conditions. In classification, when the underlying task becomes easier, such as when the decision boundary becomes more distinct or class separation increases, classification accuracy typically improves, reflecting the reduced complexity of the decision space. However, as observed in both artificial and real-world experiments, this expected behavior does not hold for existing quantifiers. Many quantification methods remain vulnerable, even when the test distribution becomes more separable. This counterintuitive behavior further exposes a critical weakness in current quantification methods. The improvements

[4] All results are available in our supplemental material repository (`https://github.com/JPOrtegaa/QuaDapt-Lequa25`)

observed with QuaDapt quantifiers directly address this issue, demonstrating that quantifiers can and should respond positively to easier test conditions.

6 Conclusion

In this paper, we present QuaDapt, a generic and adaptable framework that enables classifier-based quantifiers to be more resilient to a wide range of distribution shifts, including concept drift. While traditional quantifiers have primarily focused on addressing prior probability drift, QuaDapt enhances this capability by dynamically adjusting quantifier parameters based on the observed characteristics of test-time score distributions. This adjustment enables quantifiers to remain effective even when changes in data distribution impact classifier outputs.

Among our main contributions, we formalized the QuaDapt framework and demonstrated its effectiveness through extensive experiments on both synthetic and real-world datasets. Our findings provide the first empirical evidence that quantifiers can and should adapt to favorable distributional changes, such as increased class separability, where existing methods frequently fail.

Future research involves integrating drift detection mechanisms and systematically investigating how types of distribution shifts impact quantifier behavior, improving our understanding of quantification vulnerabilities in real-world, non-stationary settings.

Acknowledgments. We thank the Araucária Foundation (14/2024), the Innovation Agency for Sustainable Regional Development (AGEUNI - Conv. 41/2024), and the State Secretariat of Science, Technology and Higher Education of Paraná (SETI) for funding and support.

References

1. Bella, A., Ferri, C., Hernández-Orallo, J., Ramírez-Quintana, M.J.: Quantification via probability estimators. In: ICDM. pp. 737–742 (2010)
2. Casale, P., Pujol, O., Radeva, P.: Personalization and user verification in wearable systems using biometric walking patterns. Pers. Ubiquitous Comput. **16**(5), 563–580 (2012)
3. De Stefano, C., Maniaci, M., Fontanella, F., di Freca, A.S.: Reliable writer identification in medieval manuscripts through page layout features: The "avila" bible case. Engineering Applications of Artificial Intelligence **72**, 99–110 (2018)
4. Dheeru, D., Karra Taniskidou, E.: UCI machine learning repository (2017), http://archive.ics.uci.edu/ml
5. Dietterich, T.G., Kong, E.B.: Machine learning bias, statistical bias, and statistical variance of decision tree algorithms (1995)
6. Donyavi, Z., Serapiao, A.B., Batista, G.: Mc-sq and mc-mq: Ensembles for multiclass quantification. IEEE Trans. Knowl. Data Eng. (2024)
7. Esuli, A., Fabris, A., Moreo, A., Sebastiani, F.: Learning to Quantify. Springer Nature (2023)

8. Firat, A.: Unified framework for quantification (6 2016), http://arxiv.org/abs/1606.00868
9. Flovik, V.: Quantifying distribution shifts and uncertainties for enhanced model robustness in machine learning applications. arXiv preprint arXiv:2405.01978 (2024)
10. Forman, G.: Lnai 3720 - counting positives accurately despite inaccurate classification. Tech. rep. (2005)
11. Forman, G.: Quantifying counts and costs via classification. Data Min. Knowl. Discovery **17**, 164–206 (10 2008)
12. Gama, J., Medas, P., Castillo, G., Rodrigues, P.: Learning with drift detection. In: SBIA. pp. 286–295. Sao Luis, Maranhao, Brazil (2004)
13. González, P., Díez, J., Chawla, N., del Coz, J.J.: Why is quantification an interesting learning problem? Prog. Artif. Intell. **6**, 53–58 (2017)
14. González, P., Castaño, A., Chawla, N.V., Coz, J.J.D.: A review onquantification learning. ACM Comput. Surv. **50** (9 2017)
15. González, P., Moreo, A., Sebastiani, F.: Binary quantification and dataset shift: an experimental investigation. Data Min. Knowl. Discovery **38**, 1670–1712 (7 2024)
16. González-Castro, V., Alaiz-Rodríguez, R., Alegre, E.: Class distribution estimation based on the hellinger distance. Inf. Sci. **218**, 146–164 (1 2013)
17. Hassan, W., Maletzke, A., Batista, G.: Accurately quantifying a billion instances per second. In: DSAA. pp. 1–10 (10 2020)
18. Li, F., Gharakheili, H.H., Batista, G.: Quantification over time. In: ECML PKDD. pp. 282–299. Springer, Vilnius, Lithuania (2024)
19. Maletzke, A., dos Reis, D., Cherman, E., Batista, G.: On the need of class ratio insensitive drift tests for data streams. In: Second International Workshop on Learning with Imbalanced Domains: Theory and Applications. vol. 94, pp. 110–124. PMLR (10 Sep 2018)
20. Maletzke, A., Reis, D.D., Hassan, W., Batista, G.: Accurately quantifying under score variability. In: ICDM. pp. 1228–1233. IEEE (12 2021)
21. Maletzke, A., Hassan, W., Reis, D., Batista, G.: The importance of the test set size in quantification assessment. In: IJCAI. pp. 2640–2646. Yokohama, Japan (2020)
22. Maletzke, A., Reis, D.D., Cherman, E., Batista, G.: Dys: A framework for mixture models in quantification. In: AAAI. vol. 33, pp. 4552–4560 (2019)
23. Moreno-Torres, J., Raeder, T., Alaiz-Rodríguez, R., Chawla, N., Herrera, F.: A unifying view on dataset shift in classification. Pattern Recognit. **45**, 521–530 (2012)
24. Saerens, M., Latinne, P., Decaestecker, C., Saerens, M., Latinne, P., Decaestecker, C.: Adjusting the outputs of a classifier to new a priori probabilities: A simple procedure. Tech. rep. (2001)
25. Schumacher, T., Strohmaier, M., Lemmerich, F.: A comparative evaluation of quantification methods. arXiv preprint arXiv:2103.03223 (2021)
26. Storkey, A.: When Training and Test Sets Are Different: Characterizing Learning Transfer, pp. 3–28 (01 2009). https://doi.org/10.7551/mitpress/9780262170055.003.0001
27. Tsymbal, A.: The problem of concept drift: definitions and related work. Tech. rep., Department of Computer Science, Trinity College Dublin, Dublin, Ireland (2004)
28. Vanschoren, J., van Rijn, J.N., Bischl, B., Torgo, L.: Openml: Networked science in machine learning. ACM SIGKDD Explorations Newsletter **15**(2), 49–60 (2013)