An RFD-Based Approach for Concept Drift Detection in Machine Learning Systems

Loredana Caruccio, Stefano Cirillo, Giuseppe Polese and Roberto Stanzione*
{lcaruccio,scirillo,gpolese,rstanzione}@unisa.it
Department of Computer Science, University of Salerno
Fisciano, Salerno, Italy

ABSTRACT

The dynamic nature of real-world data poses several challenges for data-driven processes in different application scenarios. For instance, Machine Learning approaches completely rely on data to train predictive models. However, data can dynamically evolve over time, possibly making predictive models outdated due to concept drift, with a consequent decrease in prediction accuracy. To this end, concept drift detection techniques aim to detect such changes in order to adopt countermeasures and maintain predictive performance over time. Drift detection methods that monitor data distribution shifts play a crucial role in detecting changes without requiring feedback on model predictions. In this paper, we explore the potential of profiling metadata analysis to evaluate the impact of data evolution on model performance. Specifically, among the several profiling metadata, we focus on Relaxed Functional Dependencies (RFDs) and formalize the relationship between changes in metadata and performance trends of the predictive models over time. Moreover, we define a suite of metadata-based metrics measuring the distance between two sets of data. To validate our approach, we compared it with other distribution-based metrics on datasets with known and unknown drift. Experimental results proved that the trend of the proposed metrics is strongly correlated with the model's performance, and that they are also able to capture concept drift more effectively than traditional distribution-based approaches.

1 INTRODUCTION

Machine Learning (ML) models are increasingly relied upon for a multitude of tasks. These include personalized recommendation systems [36], natural language processing applications, such as virtual assistants [31] and sentiment analysis applications [9], and for image and speech recognition tasks [1, 4]. Moreover, ML models are deployed for more delicate and critical tasks, where inaccurate predictions can lead to potentially severe consequences. As an example, anomaly detection is crucial across various domains, such as detecting fraud [26], monitoring business processes [39], ensuring network security [25], monitoring patients [38], and predicting maintenance [15]. However, as ML models transition from the training phase to real-world deployment, they face the challenge of sustaining their effectiveness. Initially, a model may exhibit robust performance, as it captures patterns and relationships within the training data. As time progresses, the underlying assumptions may no longer hold, possibly leading to wrong recommendations.

One of the main reasons for model degradation is *concept drift*, a phenomenon that refers to changes in the underlying function that generates data. Concept drift occurs due to various factors, including evolving user preferences, external environmental changes, and other alterations in the domain dynamics. Thus, given the vast reliance on ML models we experience nowadays, it is crucial to detect changes to adopt countermeasures, such as retraining the model on a more representative, up-to-date set of data. Many methods have been introduced over the years to address this problem. Some of them monitor the model's prediction performance, while others analyze how data distribution changes, without requiring feedback on the predictions. Among these, some methods have relied on qualitative descriptors like "abrupt" and "gradual", which have been shown to have limitations due to their dependence on arbitrary boundaries [44]. This leads to the necessity of estimating the *drift magnitude* by means of quantitative measures. However, while data distribution-based approaches have the advantage of not requiring an analysis of model predictions, they are more prone to false positives [5]. Moreover, existing approaches can only capture changes in the individual attribute distributions. Thus, it could be interesting to investigate new strategies by leveraging new types of properties in the data that can support the evaluation of drifts.

Valuable properties could be extracted through *Data Profiling* techniques, which enable the discovery of a wide variety of metadata [34], including data dependencies, such as Functional Dependencies (FDs) and Relaxed Functional Dependencies (RFDs), which describe functional relationships among the dataset's attributes. A first connection between dependencies and the performance of machine learning models was discussed in [29], where these types of profiling metadata have been used to characterize the existence of a function that provides an upper bound for classification accuracy.

To the best of our knowledge, to estimate concept drift, there was no attempt to analyze metadata in terms of dependencies which can identify meaningful variations that may not be fully captured by traditional distribution-based drift detection methods. In this work, we analyze the change of data dependencies, collected in two consequent time instants, to quantify data shifts in supervised machine learning settings. In fact, unlike single-column metadata, dependencies capture the relationships between features that can be critical to model behavior. Among the variety of data dependencies, we focus on RFDs, since they are better suited for real-world scenarios involving data with inaccuracies or noise.

Specifically, we defined a suite of RFD-based metrics to quantify the divergence between the training data and a set of new samples that the model has to process. Moreover, we provide other RFD-based metrics inspired by ML measures, with the aim of capturing the performance trend of the monitored model. We evaluated the proposed metrics on datasets with *Known* and *Unknown* drift, studying how their trend is correlated to the performance of the model over time. A strong correlation would prove that analyzing the evolution of RFDs during deployment

[*] All authors contributed equally to this research.

Series ISSN: 2367-2005

10.48786/edbt.2025.66

can provide meaningful insights about concept drift, and warn about the necessity of retraining the model without evaluating its predictions. We also compared the proposed metrics with existing distribution-based measures.

To summarize, the main contributions of the proposed study are:

- A formalization of the theoretical connection between RFD evolution and the performance trend of ML models;
- A suite of RFD-based divergence measures;
- A suite of RFD-based metrics inspired by ML measures;
- An extensive evaluation process to verify the effectiveness of the proposed measures in quantifying drifts;
- A comparative evaluation of the proposed metrics with respect to other distribution-based methods.

The paper is organized as follows: in Section 2, we provide an overview of concept drift detection approaches. Section 3 provides preliminary notions, while in Section 4 we formalize the relationship between the knowledge conveyed by RFDs and concept drift. The proposed approach is described in Section 5. Finally, Section 6 shows the experimental results and Section 7 provides conclusions and discusses the future directions.

2 RELATED WORK

Concept drift detection techniques typically fall into two main categories: *performance-based* and *data distribution-based* [5]. The former relies on monitoring the model's error rate to detect potential drift. A well-known method in this category is the Drift Detection Method (DDM) [18]. It continuously monitors the probability of misclassification and its associated standard deviation. Over the years, several extensions of this approach have been proposed. For example, Baena et al. [3] proposed the Early Drift Detection Method (EDDM) by focusing on the distance between errors rather than just the error rate. This approach resulted more effective in handling gradual drifts, improving DDM especially with slower transitions. In [14], the authors defined a methodology to remove older instances linked to prolonged concepts, aiming to detect drifts earlier in the process, ultimately leading to improved model accuracy. Other proposals focus on comparing time windows for detecting drift [27, 32, 35].

On the other hand, data distribution-based approaches aim to detect significant shifts between the distributions of sampled data. In [20], the authors review distance measures for assessing numerical data shifts, recommending the Hellinger distance [24]. For instance, Ditzler and Polikar [16] used this distance to identify both gradual and abrupt changes across different data batches. Their strategy averages the Hellinger distance of individual features as drift measure and adopts an adaptive threshold to trigger warnings. Instead, Principal Component Analysis (PCA) has been conceived as a promising technique for high-dimensional data. The authors of [37] proposed a method involving the creation of a reference window filled with initial data samples. Then, PCA is applied to extract the top k principal components (PCs), and both reference and test window data are projected onto them. The final change score is the maximum Kullback-Leibler divergence among the ones of each component. This approach was extended in a more recent work [21] by employing the Hellinger distance since it provides a more robust absolute measure. A variant of this measure is employed in [2], where a hybrid framework is proposed to adapt the parameters of drift detectors with tunable thresholds based on the characteristics of a data stream. This approach considers not only the model's accuracy and distributional changes but also additional information, such as the number of observations since the last detected drift.

In addition, there exist alternative approaches, such as Discriminative Drift Detector (D3) [22] and Label Dependency Drift Detector (LD3) [23], which do not rely on labels or feedback on predictions. D3 [22] employs a discriminative classifier to detect concept drift by monitoring changes in the feature space. A sliding window maintains the temporal order of samples, and logistic regression is applied to distinguish between old and new data. Concept drift is detected when the Area Under Curve (AUC) exceeds a threshold, indicating a significant difference between new and old data. Instead, LD3 [23] is tailored for multi-label data streams, and aims to detect concept drift by using a label influence ranking method, which exploits temporal relationships between labels.

In general, performance-based approaches have the advantage of being triggered only when the model performance is affected, but they require quick feedback on the predictions made. Instead, data distribution-based approaches consider only the distributions of data samples. However, since changes in distribution may not consistently affect the model, these approaches are susceptible to false alarms [5]. Moreover, we argue that existing distance measures may suffer a loss of effectiveness in detecting drift when this does not affect the data distribution or other statistics, but only the underlying relationships between features. For this reason, the investigation of new approaches that consider other types of information/patterns within data can be considered a challenging research direction.

3 BACKGROUND NOTIONS

In this section, we introduce some preliminary notions underlying the problem statement and the proposed approach. In particular, we first provide a formal definition of *Concept Drift*, and then we introduce the definition of Relaxed Functional Dependency (RFD).

3.1 Concept Drift

Over the years, several definitions of *concept* and *concept drift* have been provided, leading to a lack of a standard definition. In this section, we comply with the definitions reported by Bayram et al. [5], representing a general and unified probabilistic view of concept.

In a supervised ML setting, each data instance is composed of a feature vector X and a target variable y. Formally, a concept drift occurs if there is a change in the joint distribution $P(X, y)$ between two time instants τ and $\tau + w$:

$$P_\tau(X, y) \neq P_{\tau+w}(X, y) \quad (1)$$

The latter can be decomposed by using the Bayesian Rule:

- $P_\tau(X|y)$ expresses the class-conditional probability density distribution, capturing the likelihood of encountering specific input data given a target label. It expresses how the features are distributed when considering a specific outcome;
- $P_\tau(y|X)$ is the posterior probability distribution of the target labels, capturing the likelihood of observing specific target labels given the features. It represents the probability of different outcomes after considering the available input data;

- $P_\tau(X)$ is the probability distribution of the input data and encapsulates the likelihood of encountering various sets of features within a dataset;
- $P_\tau(y)$ is the prior probability distribution of the target labels and expresses the likelihood of observing particular outcomes without considering the input data.

According to which of these distributions change between τ and $\tau + w$, it is possible to categorize the concept drift:

- $P_\tau(y|X) \neq P_{\tau+w}(y|X)$: a variation in the posterior probability distribution that represents a change in the model's predictions driven by new observations. It has a direct impact on the model's performance and may result in the model no longer being able to accurately predict. In this case, two types of drift can be defined: *Real concept drift*, when the change in the posterior probability might be associated with changes in $P(X)$, and *Actual drift*, when $P(X)$ remains unaltered.
- $P_\tau(X) \neq P_{\tau+w}(X)$: a variation in the probability distribution of the input data that represents common changes in data during the model deployment. Typically, changes in $P(X)$ affect model performance, since ML models struggle to adapt to unfamiliar data [13]. This kind of drift is denoted as *Covariate shift*, but it is termed *Virtual drift* when the changes do not affect the decision boundary.
- $P_\tau(Y) \neq P_{\tau+w}(Y)$: a change in the prior probability distribution that could impact the prediction performance, especially when there is a noteworthy alteration in the distribution of classes. This kind of drift is termed *Prior-probability shift* and also comprises cases when new classes are introduced or existing classes cease to exist.

To categorize concept drift, it is also possible to consider the type of shift, which can manifest in various patterns:

- *Gradual drift* represents a progressive evolution from one concept to another one over time;
- *Abrupt/Sudden drift* represents an immediate transition from a concept to another one;
- *Incremental drift* represents a slow replacement of an old concept by a new one in a continuous manner, without clearly defining a boundary between them;
- *Recurring drift* represents a phenomenon where previously observed patterns reoccur after a time interval.

In this paper, we investigate if analyzing data dependencies can provide insights in quantifying concept drifts. Since changes within data also reflect on metadata, we consider RFDs, whose definition can be found in the following section.

3.2 Relaxed Functional Dependencies (RFDs)

Data profiling tasks enable the discovery of a wide range of metadata, spanning from simple statistics related to single attributes to more complex multi-attribute metadata. Among the latter, Functional Dependencies (FDs) describe integrity constraints among relation attributes. More formally, an FD can be defined as follows:

Definition 3.1. (FD). Given a relation schema R, a Functional Dependency (FD) represents an integrity constraint that expresses a relationship between two sets of attributes X and Y, denoted as $X \rightarrow Y$ (X *implies* Y), with $X, Y \subseteq attr(R)$ and $X \cap Y = \emptyset$. An FD is satisfied on a relation instance r of R if and only if for every pair of tuples (t_1, t_2), whenever $t_1[X] = t_2[X]$, then $t_1[Y] = t_2[Y]$; where $t_i[X]$ represents the projection of the tuple t_i on the attribute set X. The attribute set $X = X_1, X_2, ..., X_h$ represents the Left Hand Side (LHS) of the FD, whereas the set $Y = Y_1, Y_2, ..., Y_k$ is the Right Hand Side (RHS).

The definition of FD has been recently extended to address challenges associated with inconsistent and inaccurate real-world data, requiring the adoption of more flexible constraints, leading to the introduction of Relaxed Functional Dependencies (RFDs). The latter can admit a limited number of violations (RFDs relaxing on the *extent*, namely RFD$_e$s) and/or the usage of similarity/distance functions as matching operators (RFDs relaxing on the *attribute comparison*, namely RFD$_c$s). In this paper, we leverage RFD$_c$s only, and in the following, we will formally introduce this type of dependency. A more general definition of RFD can be found in [11].

To enable approximate matching, RFD$_c$s leverage the concept of similarity or distance constraint. More formally, given an instance r of a relation schema R, a *constraint* ϕ, over an attribute $A \in$ *attr(R)*, is a predicate $\delta(t_i[A], t_j[A])\theta_k\varepsilon$, where δ is a similarity (or distance) function, θ_k a comparison operator, and ε a threshold. In particular, a specific similarity/distance function is applied according to the nature of the attributes (e.g., absolute difference for numerical attributes and the Levenshtein distance [30] for textual ones).

Definition 3.2. (RFD$_c$). Given a relation schema R, an RFD$_c$ φ is denoted as $X_{\Phi_1} \rightarrow Y_{\Phi_2}$, where:

- $X = X_1, X_2, ..., X_h$ and $Y = Y_1, Y_2, ..., Y_k$, with $X, Y \subseteq attr(R)$ and $X \cap Y = \emptyset$;
- $\Phi_1 = \bigwedge_{X_i \in X} \phi_i[X_i]$($\Phi_2 = \bigwedge_{Y_j \in Y} \phi_j[Y_j]$, *resp.*), with ϕ_i(ϕ_j, *resp.*) a conjunction of similarity/distance constraints on X_i(Y_j, *resp.*) and $i = 1, \ldots, h$ ($j = 1, \ldots, k$, *resp.*).

Given an instance r of R, we can state that r satisfies the RFD$_c$ φ (i.e., $r \models \varphi$) if and only if for every pair of tuples $(t_1, t_2) \in r$, if Φ_1 is true then also Φ_2 returns true.

For the sake of simplicity and without loss of generality, in the following, we only consider RFD$_c$s with a single attribute on the RHS, i.e., $X_{\Phi_1} \rightarrow A_{\phi_2}$. Moreover, the following examples will refer to constraints defined through distance functions, with a comparison operator ($\leq$) and the associated threshold.

As an example, let us consider tuples ranging from t_0 to t_6 of the snippet of a used cars dataset shown in Table 1, then an example of holding RFD$_c$ is: φ : $\mathsf{Model}_{\leq 4}, \mathsf{Year}_{\leq 1} \rightarrow \mathsf{Price}_{\leq 300}$, denoting that whenever two tuples have similar values on Model and Year, which is the case of (t_0, t_6) and (t_2, t_4), then they have a similar Price.

One of the most important characteristics of a RFD$_c$ is its *minimality*, which guarantees that the RFD$_c$ no longer holds after either (*i*) increasing one or more thresholds on the LHS constraints, (*ii*) removing an LHS attribute, or (*iii*) decreasing the RHS threshold.

Definition 3.3 (Minimal RFD$_c$). Given an instance r of a relation schema R, an RFD$_c$ φ: $X_{1 \leq \alpha_1}, \ldots, X_{h \leq \alpha_h} \rightarrow A_{\leq \beta}$ is minimal iff:

- $\rho : X_{1 \leq \alpha_1 + \varepsilon_1}, \ldots, X_{h \leq \alpha_h + \varepsilon_h} \rightarrow A_{\leq \beta}$ does not hold on r, where $\forall j = 1, \ldots, h$ then $\varepsilon_j \geq 0$, $\exists j = 1, \ldots, h$ s.t. $\varepsilon_j > 0$; and
- $\rho : X_{1 \leq \alpha_1}, \ldots, X_{i-1 \leq \alpha_{i-1}}, X_{i+1 \leq \alpha_{i+1}}, \ldots, X_{h \leq \alpha_h} \rightarrow A_{\leq \beta}$ does not hold on r, where $\exists i = 1, \ldots, h$; and
- $\rho : X_{1 \leq \alpha_1}, \ldots, X_{h \leq \alpha_h} \rightarrow A_{\leq \beta - \varepsilon_{h+1}}$ does not hold on r, where $\varepsilon_{h+1} > 0$.

Since it is difficult to apriori define proper RFD$_c$s characterizing real-world scenarios, it is necessary to exploit discovery

		Model	Year	# Owners	Price
	t_0	Hyundai i10	2016	2	6.000
-	t_1	Kia Picanto	2015	1	4.000
	t_2	Renault Clio	2018	1	8.300
	t_3	Ford Fiesta tdci	2022	1	10.500
	t_4	Renault Clio dCi	2019	1	8.600
	t_5	Fiat Panda	2019	1	8.500
	t_6	HyundaI i10 1.0	2016	1	6.250
+	t_7	Renault Clio	2019	3	6.000

Table 1: A snippet of a used car dataset.

algorithms to automatically infer them from data [12, 19, 41]. This entails searching for RFD$_c$s holding on a given dataset.

Definition 3.4 (Discovery of minimal RFD$_c$s). Given a relational schema R and an instance r of R, a discovery process consists of finding the set Σ of *all* possible RFD$_c$s $\varphi : X_{\Phi_1} \rightarrow A_{\phi_2}$ that *hold* on r (i.e., $r \vDash \Sigma$) such that $\forall \varphi \in \Sigma$, φ is also *minimal*. In other words, $\nexists\, \rho \in \Sigma$, with ρ minimal with respect to φ.

Notice that RFD$_c$ discovery algorithms deal with a problem that in the worst case is exponential in the number of attributes, since they have to browse a search space which considers all possible attribute combinations. Moreover, when the distance thresholds have to be automatically inferred the algorithm must consider all possible dispositions of distance thresholds for each attribute combination [10]. Nevertheless, thanks to specific pruning strategies, mostly based on the minimality property, it is possible to significantly reduce the search space, widely lowering discovery times.

Updating RFD$_c$s over time. The nature of real-world data is inherently dynamic, constantly evolving over time following inserts, deletions, and updates of data. Consequently, also RFD$_c$s must evolve accordingly, with respect to the type of performed operations. In what follows, we first discuss the evolution of RFD$_c$s after deletion operations and then after the insertion ones. Notice that, no specific considerations need to be done for update operations, since they can be represented as a deletion followed by an insertion one.

Let Σ be the set of RFD$_c$s holding at time τ, and Σ' the set of RFD$_c$s holding at time $\tau + 1$. A tuple deletion at time $\tau + 1$ cannot invalidate any RFD$_c$ $\varphi \in \Sigma$. Nevertheless, it could make a given $\varphi \in \Sigma$ no longer minimal, requiring the evaluation of one or more *generalizations* of φ, which can hold at time $\tau + 1$.

Definition 3.5 (Generalization of an RFD$_c$). Given a relation schema R, an instance r of R, and an RFD$_c$ φ: $X_{1 \leq \alpha_1}, \ldots, X_{h \leq \alpha_h} \rightarrow A_{\leq \beta}$ holding on r. An RFD$_c$ $\varphi' : X'_{\Phi_1} \rightarrow A_{\phi_2}$ is a generalization of φ iff:

- $\varphi' : X_{1 \leq \alpha_1 + \varepsilon_1}, \ldots, X_{h \leq \alpha_h + \varepsilon_h} \rightarrow A_{\leq \beta - \varepsilon_{h+1}}$ holds on r, where $\forall j = 1, \ldots, h+1$ then $\varepsilon_j \geq 0$, $\exists j = 1, \ldots, h+1$ s.t. $\varepsilon_j > 0$; or
- $\varphi' : X_{1 \leq \alpha_1}, \ldots, X_{i-1 \leq \alpha_{i-1}}, X_{i+1 \leq \alpha_{i+1}}, \ldots, X_{h \leq \alpha_h} \rightarrow A_{\leq \beta}$ holds on r, where $\exists i = 1, \ldots, h$; or
- $\varphi' : X_{1 \leq \alpha_1 + \varepsilon_1}, \ldots, X_{i-1 \leq \alpha_{i-1} + \varepsilon_{i-1}}, X_{i+1 \leq \alpha_{i+1} + \varepsilon_{i+1}}, \ldots, X_{h \leq \alpha_h + \varepsilon_h} \rightarrow A_{\leq \beta - \varepsilon_{h+1}}$ holds on r, where $\forall j = 1 \ldots, h+1$ then $\varepsilon_j \geq 0$, $\exists j = 1, \ldots, h+1$ s.t. $\varepsilon_j > 0$, and $\exists i = 1, \ldots, h$.

As an example, consider the tuples ranging from t_0 to t_6 shown in Table 1, and suppose that t_1 gets deleted. Thus, the RFD$_c$ φ : $\mathsf{Model}_{\leq 4}, \mathsf{Year}_{\leq 1} \rightarrow \mathsf{Price}_{\leq 300}$ is still valid, but it is no longer minimal, since φ' : $\mathsf{Year}_{\leq 1} \rightarrow \mathsf{Price}_{\leq 300}$ holds on the updated dataset.

On the other hand, a tuple insertion at time $\tau + 1$ can invalidate an RFD$_c$ $\varphi \in \Sigma$, requiring the evaluation of one or more *specializations* of φ, which can hold at time $\tau + 1$.

Definition 3.6 (Specialization of an RFD$_c$). Given a relational schema R, an instance r of R, and an RFD$_c$ φ: $X_{1 \leq \alpha_1} \ldots, X_{h \leq \alpha_h} \rightarrow A_{\leq \beta}$ holding on r. An RFD$_c$ $\varphi'' : X''_{\Phi_1} \rightarrow A_{\phi_2}$ is a specialization of φ iff:

- $\varphi'' : X_{1 \leq \alpha_1 - \varepsilon_1}, \ldots, X_{h \leq \alpha_h - \varepsilon_h} \rightarrow A_{\leq \beta + \varepsilon_{h+1}}$ holds on r, where $\forall j = 1, \ldots, h+1$ then $\varepsilon_j \geq 0$, and $\exists j = 1, \ldots, h+1$ s.t. $\varepsilon_j > 0$; or
- $\varphi'' : X_{1 \leq \alpha_1} \ldots, X_{h \leq \alpha_h}, X_{h+1 \leq \alpha_{h+1}} \rightarrow A_{\leq \beta}$ holds on r, where $\exists i = 1, \ldots, h+1$; or
- $\varphi'' : X_{1 \leq \alpha_1 - \varepsilon_1}, \ldots, X_{h \leq \alpha_h - \varepsilon_h}, X_{h+1 \leq \alpha_{h+1}} \rightarrow A_{\leq \beta + \varepsilon_{h+2}}$ holds on r,where $\forall j = 1, \ldots, h+2$ then $\varepsilon_j \geq 0$, and $\exists j = 1, \ldots, h+2$ s.t. $\varepsilon_j > 0$, and $\exists i = 1, \ldots, h+1$.

As an example, consider the tuples ranging from t_0 to t_6 shown in Table 1, and suppose that tuple t_7 is inserted. In this case, the RFD$_c$ φ : $\mathsf{Model}_{\leq 4}, \mathsf{Year}_{\leq 1} \rightarrow \mathsf{Price}_{\leq 300}$ is no longer valid, since although (t_2,t_7) and (t_4,t_7) satisfy the constraints defined on the LHS of φ, they violate the RHS constraint on the attribute Price. However, the following specialization of φ holds on the updated dataset: φ'' : $\mathsf{Model}_{\leq 4}, \mathsf{Year}_{\leq 1}, \mathsf{\#Owners}_{\leq 1} \rightarrow \mathsf{Price}_{\leq 300}$.

In this study, we formalize how to exploit these kind of RFD$_c$ evolutions with the aim of detecting concept drift. The following section will provide more details on how to characterize them.

4 RFD$_c$S AND CONCEPT DRIFT

Considering changes in data distribution represents the most popular concept drift detection technique that does not require any analysis of model predictions. These approaches try to establish if more recent data comes from a different distribution with respect to the older one, risking overlooking drifts that are not immediately evident in the overall data distribution. This can be due to the fact that concept drift does not always produce changes in the statistical properties of data, but rather shifts in its underlying relationships. To this end, we investigate whether, by capturing the evolution of relationships through metrics based on the analysis of RFD$_c$ evolution, it is possible to achieve any findings for determining concept drift. As discussed in Section 3.2, RFD$_c$s may evolve as generalizations, specializations, or may be invalidated (i.e., they do not hold on the updated data and are neither specialized nor generalized). These evolutions can be evaluated to quantify the divergence between two sets of RFD$_c$s.

Let us consider the sets Σ and Σ' of RFD$_c$s holding on a relation instance r of R in two given time instants τ and $\tau + 1$, respectively. The analysis of how RFD$_c$s change between the two time instants can be accomplished in two different perspectives: evaluating how much Σ is changed with respect to Σ', and vice versa. In what follows, we formalize the characterization of all possible changes on the sets of holding RFD$_c$s according to these two scenarios.

Definition 4.1 (Shift from Σ to Σ'). To quantify the degree of divergence between Σ and Σ', it is necessary to evaluate each RFD$_c$ $\varphi \in \Sigma$ to verify if φ is somehow related to any RFD$_c$ $\varphi' \in \Sigma'$. Specifically, $\forall \varphi \in \Sigma$:

- φ can also belong to Σ';
- φ can be specialized by at least one $\varphi' \in \Sigma'$;
- φ can neither belong to Σ' nor be specialized by any $\varphi' \in \Sigma'$, meaning that φ has been invalidated.

As an example, let us consider the sets of RFD$_c$s Σ and Σ' provided in Table 2, which are discovered at time τ and $\tau + 1$, respectively. By examining the RFD$_c$s in Σ, we can quantify the shift as follows:

- the RFD$_c$ φ_2 does not change, since it also belongs to Σ';

φ	Σ
φ_0	Model$_{\leq 0}$, #Owners$_{\leq 1}$ $\rightarrow$ Price$_{\leq 300}$
φ_1	Model$_{\leq 1}$, Year$_{\leq 0}$ $\rightarrow$ Price$_{\leq 300}$
φ_2	Year$_{\leq 0}$, Price$_{\leq 300}$ $\rightarrow$ Model$_{\leq 0}$
φ_3	Model$_{\leq 1}$, Price$_{\leq 300}$ $\rightarrow$ Year$_{\leq 1}$
φ_4	Model$_{\leq 0}$, #Owners$_{\leq 1}$ $\rightarrow$ Year$_{\leq 1}$
φ_5	Price$_{\leq 300}$ $\rightarrow$ #Owners$_{\leq 0}$
φ_6	Year$_{\leq 0}$ $\rightarrow$ #Owners$_{\leq 0}$

φ	Σ'
φ'_0	Model$_{\leq 0}$, #Owners$_{\leq 1}$, Year$_{\leq 0}$, $\rightarrow$ Price$_{\leq 300}$
φ'_1	Year$_{\leq 0}$, Price$_{\leq 300}$ $\rightarrow$ Model$_{\leq 0}$
φ'_2	Model$_{\leq 1}$, Price$_{\leq 300}$, #Owners$_{\leq 1}$, $\rightarrow$ Year$_{\leq 1}$
φ'_3	Price$_{\leq 300}$ $\rightarrow$ #Owners$_{\leq 2}$
φ'_4	Year$_{\leq 3}$, #Owners$_{\leq 1}$ $\rightarrow$ Model$_{\leq 2}$

Table 2: An example of RFD$_c$ sets holding at two time instants τ and $\tau + 1$.

- 5 RFD$_c$s are specialized in Σ'. In particular, φ_0 and φ_1 are specialized by φ'_0, φ_3 and φ_4 by φ'_2, and φ_5 by φ'_3;
- the RFD$_c$ φ_6 is invalidated, since it does not belong to Σ' and there is no RFD$_c$ in Σ' that specializes it.

In other words, this type of analysis allows to evaluate the change of an older RFD$_c$ set compared to a more recent one. Conversely, as mentioned, it is also possible to perform an analysis evaluating the change of a newer RFD$_c$ set compared to an older one.

Definition 4.2 (Shift from Σ' to Σ). To quantify the degree of divergence between Σ' and Σ, it is necessary to evaluate each RFD$_c$ $\varphi' \in \Sigma'$ to verify if φ' is somehow related to any RFD$_c$ $\varphi \in \Sigma$. Specifically, $\forall \varphi' \in \Sigma'$:

- φ' can also belong to Σ;
- φ' can be generalized by at least one $\varphi \in \Sigma$;
- φ' can neither belong to Σ nor be generalized by any $\varphi \in \Sigma$, meaning that φ' is a new RFD$_c$.

As an example, consider the sets of RFD$_c$s Σ and Σ' in Table 2. By examining the RFD$_c$s in Σ', we can quantify its shift as follows:

- the RFD$_c$ φ'_1 does not change, since it also belongs to Σ;
- 3 RFD$_c$s are generalized in Σ. In particular, φ'_0 is generalized by φ_0 and φ_1; φ'_2 by φ_3 and φ_4; and φ'_3 by φ_5;
- the RFD$_c$ φ'_4 is a new RFD$_c$, since it does not belong to Σ and there is no RFD$_c$ in Σ that generalizes it.

From this characterization of possible changes between Σ and Σ', a more fine-grained analysis can be introduced by considering further properties of generalizations and specializations. We will present some of these characterizations in the following section.

5 THE PROPOSED APPROACH

In what follows, we describe the proposed concept drift detection methodology. In particular, we first discuss the reason why analyzing shifts in RFD$_c$s can provide effective support in detecting concept drift. After a general overview of the proposed approach, we describe it by detailing each involved step.

5.1 Exploiting RFD$_c$s for concept drift detection

As further contribution, we introduce an approach tailored for supporting supervised ML models by detecting concept drift. This is accomplished through analyses of the shift in terms of RFD$_c$s during the model deployment, in order to warn about the necessity of retraining the model if the shift is severe. Since concept drift can occur only for some of the target labels, our approach evaluates the shift for each class, so it can provide more detailed insights into which specific classes are affected by the drift.

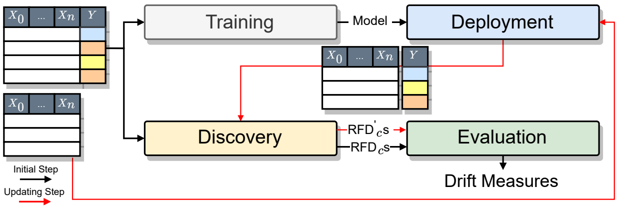

Figure 1: A general overview of the proposed approach.

Figure 1 shows an overview of the proposed approach. As we can see, it is based on two main steps, whose operation flows are highlighted in two different colors (i.e., black and red arrows). In particular, in the training phase, the model that has to be monitored over time is trained on the available data. At the same time, a discovery process is performed on the same data, through which it is possible to extract the set of holding RFD$_c$s for each target label. In the next step, the trained model is deployed and starts to make predictions on incoming data.

As previously discussed, the model's performance can degrade over time due to a drift occurring in new data. However, since there is no knowledge about the predictions made, our approach entails conducting periodical checks to verify whether the data at hand has significantly changed. To accomplish this, a new RFD$_c$ discovery process is performed on an updated dataset, which consists of the concatenation of the training data and the new instances predicted by the model. Then, for each class, the original set of RFD$_c$s holding on the training data is compared with the updated set of RFD$_c$s. To perform this comparison, we leverage several RFD-based metrics that are defined later. If through these metrics a significant shift is highlighted, it may be necessary to retrain the model.

5.2 Collecting Meaningful RFD$_c$s

Figure 2 shows in detail the proposed approach. By following one of the main goals of it, sets of minimal RFD$_c$s are collected in different steps of the whole process, i.e., the *initial* and the *updating* one. Nevertheless, independently from them, the proposed approach underlies three main phases for collecting meaningful RFD$_c$s: i) Preprocessing, ii) RFD$_c$ Discovery, and iii) RFD$_c$ Filtering; as highlighted by the yellow box in Figure 2.

5.2.1 *Preprocessing.*

The preprocessing phase aims to prepare the dataset in input for the RFD$_c$ discovery. The main operations performed in this phase are the selection of the most relevant features, the organization of data into equivalence classes, and the splitting of data. The first two operations aim to reduce the number of features and the variability in the data, allowing for RFD$_c$ discovery processes to quickly focus on the most informative features, by also avoiding data having a too fine-grained representation. Specifically, for the first operation, we leverage mutual information-based feature selection [28] to maintain only the most relevant attributes for the follow-up analyses. This technique was elected as distinguished technique for its ability in capturing any type of relationship [17], and for its robustness to noise and data transformations, yielding effective selection of features [33]. The second operation is applied to attributes with

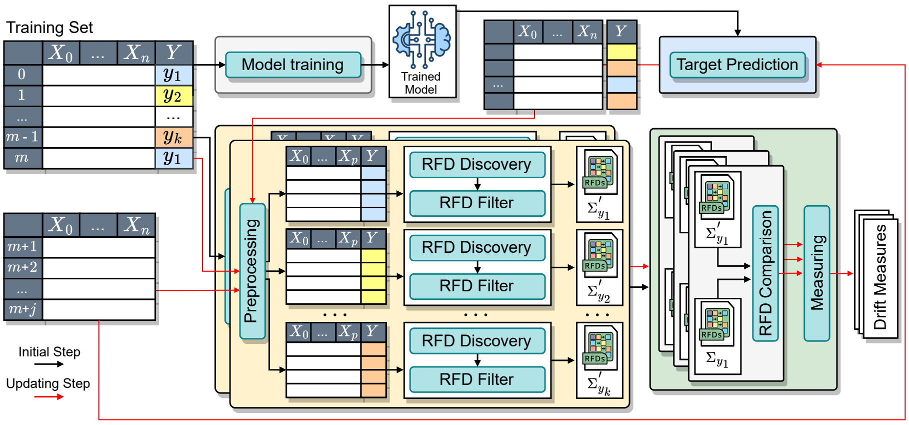

Figure 2: An overview of the initialization and updating steps underlying the proposed approach.

a high variability of values, which are arranged into equivalence classes, ensuring that similar values are assigned to the same class [12] and replaced with the identifier of such an equivalence class. In the third operation, the dataset is split according to the k target labels. Thus, the output of this step consists of k subsets, each containing instances belonging to a specific class.

5.2.2 RFD$_c$ Discovery. After the preprocessing phase, each of the k subsets is given as input to an RFD$_c$ discovery algorithm. In particular, for this approach, we leverage the DOMINO algorithm [10], which has the peculiarity of inferring by itself the distance constraints for each attribute. This feature is useful when the application domain is not well known and there is uncertainty about the distance constraints to apply. However, any other RFD$_c$ discovery algorithm can be chosen based on the characteristics of the application domain. The output of this phase consists of k sets of RFD$_c$s, i.e., Σ_{y_i}, $i = 1, 2, \ldots, k$, where k is the number of target labels.

5.2.3 RFD$_c$ Filtering. For a given dataset, the presence of holding RFD$_c$s can scale to the order of thousands or even more. In order to perform the subsequent comparison steps, in this phase we filter discovered dependencies, aiming at retaining, for each target label, only its most representative RFD$_c$s, i.e., those that distinguish it most from the others. To perform initial filtering of the original dependencies, we remove from each set Σ_{y_i} all the RFD$_c$s that are also present in the other sets Σ_{y_j}, with $j = 1, 2, \ldots, k$ and $i \neq j$. According to this, we can define Σ_{y_i} as:

$$\Sigma_{y_i} = \Sigma_{y_i} \setminus \bigcup_{i,j=1,2,\ldots,k \,\wedge\, i \neq j} \Sigma_{y_j} \tag{2}$$

Thus, the resulting k RFD$_c$ sets contain, for each class, only its unique dependencies. After that, we further filter RFD$_c$s by leveraging the concept of minimality. In particular, for each set Σ_{y_i}, we maintain only RFD$_c$s that are minimal with respect to all the RFD$_c$s belonging to other sets Σ_{y_j}, with $j \neq i$. This ensures that the RFD$_c$s of each class are unique and not related to those discovered for the other classes. The output of this phase consists of the updated k RFD$_c$ sets, each encapsulating the most representative dependencies for its respective target class.

As an example, let us consider a scenario in which two target labels are involved, namely y_i and y_j, respectively. Thus, supposing that after the discovery process at given time τ, the following resulting RFD$_c$s are provided:

- $\Sigma_{y_i} = \Sigma \cup \{\varphi_7\}$, where
  Σ refers to the set of RFD$_c$s shown in Table 2 and
  φ_7: $\text{Year}_{\leq 2}, \#\text{Owners}_{\leq 1} \longrightarrow \text{Model}_{\leq 3}$; and
- $\Sigma_{y_j} = \{\varphi_7, \varphi_8, \varphi_9\}$, where
  φ_8: $\text{Model}_{\leq 2}, \text{Price}_{\leq 300} \longrightarrow \#\text{Owners}_{\leq 0}$ and
  φ_9: $\text{Year}_{\leq 3} \longrightarrow \text{Model}_{\leq 2}$.

Notice that the RFD$_c$ φ_7 is shared between the two sets Σ_{y_i} and Σ_{y_j} and that $\varphi_8 \in \Sigma_{y_j}$ is not minimal with respect to $\varphi_5 \in \Sigma_{y_i}$. Consequently, the sets Σ_{y_i} and Σ_{y_j} will become $\Sigma_{y_i} = \{\varphi_0, \varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_5, \varphi_6\}$ and $\Sigma_{y_j} = \{\varphi_9\}$ after the application of the filtering strategy.

5.3 Evaluating Drift through RFD$_c$s

As mentioned in Section 5.1, our approach entails the assessment of updated data to verify whether significant changes occur also in terms of RFD$_c$s. This detection serves as a trigger to retrain the model and enhance its adaptability to changing data dynamics. In particular, during the deployment, the model trained in the previous step starts making predictions on incoming data, and these predicted instances are incrementally integrated into the original training dataset. Thus, the new data (comprised of the original training data with true labels and the newly predicted instances) is reintroduced to the discovery module where the same sequence of operations described in Section 5.2 (i.e., preprocessing, discovery, and RFD$_c$ filtering) is performed.

Starting from the sets of RFD$_c$s resulting from the updated sets of data, the drift evaluation can be performed. Specifically, in this step, the proposed approach considers both the Σ_{y_i} and Σ'_{y_i} sets of RFD$_c$s, with $i = 1, 2, \ldots, k$ and k the number of target labels (see Section 5.2). Then, it consists of two main phases that underlie the whole evaluation process: i) RFD$_c$ Comparison and ii) Measuring; as highlighted by the green box in Figure 2.

5.3.1 RFD$_c$ Comparison. This phase takes care of comparing, for each target label, the original and updated RFD$_c$ sets, according to type of shifts among sets of RFD$_c$s (see Section 4). Notice that, in what follows we describe the comparison methodology between two sets of RFD$_c$s, since it can be easily generalized to the comparison of all sets of RFD$_c$s in Σ_{y_i} and Σ'_{y_i} with $i = 1, 2, \ldots, k$.

The comparison between Σ_{y_i} and Σ'_{y_i} provides different interpretations according to the direction followed during the process. In fact, the shift of RFD$_c$s can be performed from Σ_{y_i} and Σ'_{y_i} (see Definition 4.1) and vice versa (see Definition 4.2), yielding

Notation	Definition
Σ_{y_i}	RFD$_c$s holding on the training instances with target class y_i
Σ'_{y_i}	RFD$_c$s holding on the training instances and the new predicted ones with target class y_i
Imm	Number of RFD$_c$s that belongs both to Σ_{y_i} and to Σ'_{y_i}
Inv	Number of RFD$_c$s in Σy_i that are not present in any form in $\Sigma' y_i$
$Spec$	Number of RFD$_c$s in Σ_{y_i} that are specialized in Σ'_{y_i}
$Spec_{\supset}$	Number of RFD$_c$s in Σ_{y_i} that are specialized in Σ'_{y_i} by an RFD$_c$ with the same thresholds on the attributes in common
$Spec_{\supset\Rightarrow}$	Number of RFD$_c$s in Σ_{y_i} that are specialized in Σ'_{y_i} by an RFD$_c$ with lower or equal thresholds on the attributes in common on the LHS, and higher threshold on the RHS
$Spec_{\supset_1}$	Number of RFD$_c$s in Σ_{y_i} that are specialized in Σ'_{y_i} by an RFD$_c$s with only an additional attribute on the LHS
$Spec_{eq}$	Number of RFD$_c$s in Σ_{y_i} that are specialized in Σ'_{y_i} by an RFD$_c$ with the same attributes
$Spec_{eq\Leftarrow}$	Number of RFD$_c$s in Σ_{y_i} that are specialized in Σ'_{y_i} by an RFD$_c$ with the same attributes, with the same threshold on the RHS and lower or equal thresholds on the LHS (with at least one lower)
$Spec_{eq\Rightarrow}$	Number of RFD$_c$s in Σ_{y_i} that are specialized in Σ'_{y_i} by an RFD$_c$ with the same attributes, with the same thresholds on the LHS and greater or equal threshold on the RHS (with at least one greater)
$Spec_{eq\Leftrightarrow}$	Number of RFD$_c$s in Σ_{y_i} that are specialized in Σ'_{y_i} by an RFD$_c$ with the same attributes, with greater threshold on the RHS and lower or equal thresholds on the LHS (with at least one lower)
New	Number of RFD$_c$s in Σ'_{y_i} that are not derived from any RFD$_c$ in Σ_{y_i}
Gen	Number of RFD$_c$s in Σ'_{y_i} that are generalized in Σ_{y_i} in any form
$Gen_{\subset}$	Number of RFD$_c$s in Σ'_{y_i} that are generalized in Σ_{y_i} by an RFD$_c$ with the same thresholds on the attributes in common
$Gen_{\subset\Rightarrow}$	Number of RFD$_c$s in Σ'_{y_i} that are generalized in Σ_{y_i} by an RFD$_c$ with the same threshold on the LHS and lower or equal thresholds on the RHS on the attributes in common (with at least one lower)
$Gen_{\subset_1}$	Number of RFD$_c$s in Σ'_{y_i} that are generalized in Σ_{y_i} by an RFD$_c$ with only one attribute less on the LHS
Gen_{eq}	Number of RFD$_c$s in Σ'_{y_i} that are generalized in Σ_{y_i} by an RFD$_c$ with the same attributes involved
$Gen_{eq\Leftarrow}$	Number of RFD$_c$s in Σ'_{y_i} that are generalized in Σ_{y_i} by an RFD$_c$ with the same attributes, with the same threshold on the RHS and greater or equal thresholds on the LHS (with at least one greater)
$Gen_{eq\Rightarrow}$	Number of RFD$_c$s in Σ'_{y_i} that are generalized in Σ_{y_i} by an RFD$_c$ with the same attributes, the same thresholds on the LHS, and lower or equal thresholds on the RHS (with at least one lower)
$Gen_{eq\Leftrightarrow}$	Number of RFD$_c$s in Σ'_{y_i} that are generalized in Σ_{y_i}

Table 3: Reference table for notations.

Metric	Definition
D_1	$\frac{New + ((Gen - Gen_{eq}) \cdot 0.5) + (Gen_{eq} \cdot 0.05)}{\lvert\Sigma'_{y_i}\rvert}$
D_2	$\frac{New + ((Gen - Gen_{eq}) \cdot 0.5) + (Gen_{eq} \cdot 0.05) + Inv}{\lvert\Sigma_{y_i} \cup \Sigma'_{y_i}\rvert}$
D_3	$\frac{New + Inv}{\lvert\Sigma_{y_i} \cup \Sigma'_{y_i}\rvert}$
D_4	$\frac{\begin{array}{c} New + (Gen_{\subset_1} \cdot 0.25) + ((Gen - \\ - Gen_{\subset_1} - Gen_{eq}) \cdot 0.5) + (Gen_{eq} \cdot 0.05) \end{array}}{\lvert\Sigma_{y_i} \cup \Sigma'_{y_i}\rvert}$
D_5	$\frac{Inv + ((Spec - Spec_{eq}) \cdot 0.5) + (Spec_{eq} \cdot 0.05)}{\lvert\Sigma_{y_i}\rvert}$
D_6	$\frac{New + (Gen \cdot 0.5) + ((Gen_{eq\Rightarrow} + Gen_{eq\Leftarrow} + Gen_{eq\Leftrightarrow}) \cdot 0.05)}{\lvert\Sigma_{y_i} \cup \Sigma'_{y_i}\rvert}$
D_7	$\frac{\begin{array}{c} Inv + (Spec_1 \cdot 0.25) + ((Spec - Spec_{\supset_1} - \\ - Spec_{eq}) \cdot 0.5) + (Spec_{eq} \cdot 0.05) \end{array}}{\lvert\Sigma_{y_i}\rvert}$
D_8	$\frac{\begin{array}{c} New + Inv + ((Gen_{eq} - (Gen_{eq\Rightarrow} + Gen_{eq\Leftarrow} + \\ + Gen_{eq\Leftrightarrow}) \cdot 0.5) + ((Gen_{eq\Rightarrow} + Gen_{eq\Leftarrow} + Gen_{eq\Leftrightarrow}) \cdot 0.05) \end{array}}{\lvert\Sigma_{y_i} \cup \Sigma'_{y_i}\rvert}$
D_9	$\frac{\begin{array}{c} New + Inv + ((Gen - Gen_{\subset_1} - Gen_{eq}) \cdot 0.5) + (Gen_{\subset_1} \cdot 0.25) + (Gen_{eq} \cdot \\ \cdot 0.05) + ((Spec - Spec_{\supset_1} - Spec_{eq}) \cdot 0.5) + (Spec_{\supset_1} \cdot 0.25) + (Spec_{eq} \cdot 0.05) \end{array}}{\lvert\Sigma_{y_i} \cup \Sigma'_{y_i}\rvert}$
D_{10}	$\frac{\begin{array}{c} Inv + (Spec_{\supset_1} \cdot 0.1) + ((Spec - Spec_{\supset_1} - \\ - Spec_{eq}) \cdot 0.2) + (Spec_{eq} \cdot 0.02) \end{array}}{\lvert\Sigma_{y_i}\rvert}$
D_{11}	$\frac{New + Inv + (Spec \cdot 0.1) + (Gen \cdot 0.1)}{\lvert\Sigma_{y_i} \cup \Sigma'_{y_i}\rvert}$
D_{12}	$\frac{Inv + ((Spec - Spec_{eq}) \cdot 0.3) + (Spec_{eq} \cdot 0.02)}{\lvert\Sigma_{y_i}\rvert}$

Table 4: RFD-based divergences.

different types of changes between the two sets of RFD$_c$s. Nevertheless, it is possible to quantify the occurrence of specific types of changes involved during the comparison process as shown in Table 3.

Specifically, independently from the direction, there can be a certain number of RFD$_c$s that appear equal in both sets, namely *Imm*. Instead, if the comparison is performed from Σ_{y_i} to Σ'_{y_i}, then it is possible to quantify the number of RFD$_c$s in Σ_{y_i} that are:

- generically specialized in Σ'_{y_i}, namely *Spec*;
- specialized in Σ'_{y_i} by adding attributes on the LHS, namely $Spec_{\supset}$, $Spec_{\supset\Rightarrow}$, $Spec_{\supset_1}$;
- specialized in Σ'_{y_i} by varying thresholds only, namely $Spec_{eq}$, $Spec_{eq\Leftarrow}$, $Spec_{eq\Rightarrow}$, $Spec_{eq\Leftrightarrow}$;
- invalidated, i.e., neither present nor specialized in Σ'_{y_i}, namely *Inv*.

Notice that, details about the different criteria of specializations/generalizations are provided in Table 3.

As an example, let us consider the two sets of RFD$_c$s Σ and Σ' shown in Table 2, which can be denoted as Σ_{y_i} and Σ'_{y_i} since they are associated to a single label. Thus, by performing a comparison from Σ_{y_i} to Σ'_{y_i}, it is possible to say that there is just one *Imm* RFD$_c$, i.e., φ_2 and one *Inv* RFD$_c$, i.e., φ_6. Moreover, among the five *Spec* RFD$_c$s: φ_0, φ_1, φ_3 φ_4, and φ_5; only the latter, compared with φ'_3, satisfies the $Spec_{eq}$ and the $Spec_{eq\Rightarrow}$ criteria since the specialization is driven by a simple variation of the RHS threshold. Instead, the other four RFD$_c$s all satisfy the $Spec_{\supset_1}$ criterion since just one attribute is added on the LHS (i.e., φ_0 and φ_1 with φ'_0, and φ_3, φ_4 with φ'_2). Moreover, φ_0 and φ_3 also satisfy $Spec_{\supset}$ since they maintain the same thresholds on common attributes when compared with φ'_0 and φ'_2, respectively.

On the other hand, if the comparison is performed from Σ'_{y_i} to Σ_{y_i}, then it is possible to quantify the RFD$_c$s in Σ'_{y_i} that are:

- generically generalized in Σ_{y_i}, namely *Gen*;
- generalized in Σ_{y_i} by removing attributes on the LHS, namely $Gen_{\subset}$, $Gen_{\subset\Rightarrow}$, $Gen_{\subset_1}$;
- generalized in Σ_{y_i} by varying thresholds only, namely Gen_{eq}, $Gen_{eq\Leftarrow}$, $Gen_{eq\Rightarrow}$, $Gen_{eq\Leftrightarrow}$;
- new, i.e., neither present nor generalized in Σ_{y_i}, namely *New*.

As an example, let us consider the same sets of RFD$_c$s involved in the previous example. By performing a comparison from Σ'_{y_i} to Σ_{y_i}, it is possible to say that there is just one *Imm* RFD$_c$, i.e., φ'_1 and one *New* RFD$_c$, i.e., φ'_4. Moreover, among the three *Gen* RFD$_c$s: φ'_0, φ'_2, φ'_3; only the latter, compared with φ_5, satisfies the Gen_{eq} and the $Gen_{eq\Rightarrow}$ criteria since the generalization is driven by a simple variation of the RHS threshold. Instead, the other two RFD$_c$s all satisfy the $Gen_{\subset_1}$ criterion since just one attribute does not appear on the LHS (i.e., φ'_0 with φ_0 and φ_1, and φ'_2 with φ_3 and φ_4). Moreover, φ'_0 and φ'_2 also satisfy $Gen_{\subset}$ criterion since they maintain the same thresholds on common attributes when compared with φ_0 and φ_3, respectively.

To summarize, many criteria can be used to evaluate the shift between two sets of RFD$_c$s, according to possible invalidation of some RFD$_c$s, the generation of new RFD$_c$s, and/or different types of specializations/generalizations. In particular, the latter can allow the evaluation at different levels of granularity. Overall, the quantitative information provided by the several comparison criteria can be used to define different metrics to measure a possible drift into data, which is reflected in the variations on holding RFD$_c$s as described in the next section.

5.3.2 *Measuring.*

From the comparison phase, all the terms shown in Table 3 are obtained. Starting from them, the proposed approach can measure the shift in terms of RFD$_c$s according to a suite of proposed RFD-based metrics. Some of them evaluate the magnitude of the change of Σ_{y_i} with respect to Σ'_{y_i}, while others consider the opposite direction, assessing the changes in Σ'_{y_i} with respect to Σ_{y_i}. Specifically, we defined two categories of metrics: the ones in the first category aim to quantify the *divergence* between two sets of RFD$_c$s, while those in the second category are inspired by machine learning, following a *confusion matrix-based* evaluation.

Metric	True Positives	False Positives	False Negatives
CF_1	Imm	New	Inv
CF_2	$Imm + \sum_{\forall G \in Gen_{eq^*}} G$	$New + (Gen - \sum_{\forall G \in Gen_{eq^*}} G)$	Inv
CF_3	$Imm + \sum_{\forall G \in Gen_{eq^*}} G$	New	Inv
CF_4	Imm	$New + Gen$	Inv
CF_5	Imm	$Inv + Spec$	New
CF_6	Imm	$Inv + Spec$	$New + Gen$
CF_7	$Imm + Gen$	$Inv + Spec$	New

^ $Gen_{eq}* = \{Gen_{eq}, Gen_{eq_{\Rightarrow}}, Gen_{eq_{\Leftarrow}}, Gen_{eq_{\Leftrightarrow}}\}$.

Table 5: Metrics inspired by confusion matrix items.

Table 4 presents the suite of *divergence* metrics, which range from simpler metrics to more refined ones. Specifically, D_5, D_7, D_{10}, and D_{12} are based on the characterizations of the shift from Σ_{y_i} to Σ'_{y_i}, while D_1 consider the shift from Σ'_{y_i} to Σ_{y_i}. Instead, the remaining metrics consider both perspectives, providing a single divergence value that summarizes the change in both sets. As shown in Table 4, the proposed metrics leverage coefficients to weight different types of RFD$_c$ evolution, assigning greater importance to more substantial changes (e.g., invalidations and new RFD$_c$s) while attributing a lower contribution to moderate ones (e.g., specializations and generalizations), aiming at accurately estimating the severity of changes among the RFD$_c$ sets. Notice that all the defined divergence metrics are normalized with respect to the total number of RFD$_c$s involved in the specific evaluation.

As an example, let us consider the two sets of RFD$_c$s Σ and Σ' shown in Table 2, which can also be denoted as Σ_{y_i} and Σ'_{y_i} since they are associated to a single label. Thus, according to Table 4, it is possible to apply the metric D_5 on the considered scenario to measure the divergence from Σ_{y_i} to Σ'_{y_i}:

$$D_5 = \frac{Inv + ((Spec - Spec_{eq}) \times 0.5) + (Spec_{eq} \times 0.05)}{|\Sigma_{y_i}|} =$$

$$= \frac{1+((5-1)\times 0.5)+(1\times 0.05)}{7} = \frac{1+2+0.05}{7} = 0.44.$$

The second category of metrics is inspired by the confusion matrix, which is commonly employed for ML evaluation. In particular, we adapted the concepts of True\False Positives and True\False Negatives to evaluate the changes among the compared sets of RFD$_c$s. Our aim is to investigate whether metrics derived from our (re-)interpretation of the confusion matrix align with the trends observed in the actual model.

Table 5 shows the seven metrics we defined. Specifically, some of these interpretations (i.e., CF_1, CF_2, CF_3, CF_4) evaluate the degree of change from Σ_{y_i} to Σ'_{y_i}, while others (i.e., CF_5, CF_6, CF_7) consider the opposite perspective, describing the change from Σ'_{y_i} to Σ_{y_i}. To clarify this interpretation, let us consider the first metric (i.e., CF_1), through which we can identify *True Positives* as the number of RFD$_c$s that were in Σ_{y_i} and that are still in Σ'_{y_i}. In other words, these are the RFD$_c$s that we expected to have and that are correctly in Σ'_{y_i}. By following the same reasoning, we can consider *False Negatives* as the RFD$_c$s that were in Σ_{y_i} but not in Σ'_{y_i}. Instead, *False Positives* represent RFD$_c$s that were not in Σ_{y_i} but in Σ'_{y_i} (e.g., new RFD$_c$s). *True Negatives* are always equal to zero, since they represent RFD$_c$s not included in both Σ_{y_i} and Σ'_{y_i}.

Consequently, we considered the *F1-Measure*, *Precision*, and *Recall* metrics computed through this adapted interpretation of the confusion matrix. In general, lower values for *Precision*, *Recall*, and *F1-Measure* indicate a larger change between the two sets of RFD$_c$s. This is due to the fact that *False Positives* and *False Negatives* are associated with the possible evolutions of RFD$_c$s, while *True Positives* are associated with RFD$_c$s that do not change or slightly evolve with respect to the original ones.

As an example, let us consider the two sets of RFD$_c$s Σ and Σ' shown in Table 2, which can also be denoted as Σ_{y_i} and Σ'_{y_i} since they are associated to a single label. Thus, according to Table 5, it is possible to interpret confusion matrix items as defined by the metric CF_5 on the considered scenario: $TruePositives = Imm = 1$; $FalsePositives = (Inv + Spec) = (1 + 5)$; and $FalseNegatives = New = 1$. Thus, starting from these items it is possible to compute the *Precision*, the *Recall*, and the *F1-Measure*.

6 EXPERIMENTAL EVALUATION

In this section, we present the evaluation performed on the proposed RFD-based metrics, aiming at demonstrating their effectiveness in quantifying possible drifts within data. To this end, we investigate whether the trends of the RFD-based metrics are more correlated to the performance trend of the model with respect to other related approaches. As a matter of fact, a higher correlation would mean that the proposed metrics are better able to capture changes within a dataset, leading to reliable insights into whether a model's performance is declining or not. Different scenarios were analyzed to demonstrate that (i) RFD-based metrics provide more reliable estimations than baseline distances and that (ii) RFD-based metrics also capture drifts within data when it does not affect the value distribution but the relationships among attributes.

6.1 Baseline approaches

To compare the proposed approach with related techniques, we leveraged FROUROS [40], a Python library that provides a wide variety of algorithms for drift detection. Specifically, since our approach falls into the category of data distribution-based approaches, we considered available *Data Drift* measures. Among these, we considered approaches that quantify the shift in the data and provide a value within a bounded range, as in the case of the proposed metrics. The first baseline method we considered is the Hellinger distance [24], recommended by [20] and leveraged by several approaches in the literature [16, 21]. Instead, a second baseline method we considered is *HiNormalizedComplement* [42]. Since these measures quantify drift for only a single attribute, for both we employed two aggregation strategies to obtain a single distance between two batches of data: (i) by using the average of all distances [16] and (ii) by using the maximum between all distances [37]. In what follows, for the *Hellinger* (*HiNormalizedComplement*, resp.) distance, we refer to the first strategy as He_{mean} (Hi_{mean}, resp.) and to the second strategy as He_{max} (Hi_{max}, resp.).

6.2 Experimental settings

The experimental evaluation has been performed in two different phases. In the first one, we considered datasets with *Known Drift* to compare the effectiveness of all the considered approaches and to determine the best metrics among the proposed ones. Thereafter, we employ the latter on datasets with *Unknown Drift* to simulate a real-world scenario. In what follows, we describe the datasets used in our evaluation (see Table 6), by also providing details about the evaluation process.

Datasets with known drift. The datasets considered in scenario can be divided into two groups. The first one, namely *Statistical Drift*, contains 9 configuration of the *Followers* dataset obtained from a repository that allowed us to mixing up normal data and data affected by gradual or abrupt drift over a variable number of columns. The second group, namely *Attribute-relationship Drift*, considers three classification datasets (i.e., *Recruitment*, *Age*, and *Forest CovType*) and their drifted version, which have been syntetically generated by shuffling the values of their columns independently (i.e., without mixing values between different columns). This ensures that the distribution of each attribute remains unchanged, while the underlying relationships between different attributes are altered. Specifically, for each dataset we shuffled the values of three columns (🟢), all columns except the target attribute (🔺), and half of the columns for one portion of data and all columns for the remaining one (🔶). The latter generation method aims to simulate a more gradual drift.

Datasets with unknown drift. For this scenario, we considered 5 datasets: *Bankrupt*, *Event Logs*, *Cleveland*, *KC2*, and *Heart-Statlog*. *Bankrupt* contains financial data collected from 1999 to 2009. The *Event Logs* [43] dataset contains data used for training a Meta-Learning system to recommend the best anomaly detection algorithm. It is composed of 365 instances, which comprises the 168 used by the authors and other synthetic ones. The *Cleveland* dataset contains data about heart diseases in patients, while the *KC2* dataset contains software metrics extracted from source code to predict software defects. Finally, the *Heart-Statlog* dataset contains medical attributes related to heart health.

As shown in Table 6, for all datasets configurations, we randomly sampled a certain number of rows in order to vary the data within each configuration. Instead, we used all samples for smaller datasets (i.e., datasets with ID from 21 to 24). The number of classes reported in Table 6 is referred to those appearing in the sampled data. Notice that, although our approach can be applied on any type of data, we faced the necessity of considering datasets with only numerical features due to the requirements of the baseline approaches.

Evaluation Process. All the experimental sessions we performed involved a partitioning process to split datasets into four batches, corresponding to the 25%, 45%, 70%, and 100% of their size, respectively. Thus, the first batch contains tuples from 0% to 25%, the second one contains tuples from 25% to 45%, and so forth. The first batch is also used for training a Random Forest model, which is deployed for making predictions over the other batches. For datasets with *Known Drift*, we sampled normal data for training and for the the first test batch, whereas we sampled drifted data for the successive test batches. Thus, we performed the preprocessing step on the training dataset by applying mutual information-based feature selection to only consider the most relevant features (see *Sel. Features* column in Table 6). Then, for each class we performed a RFD_c discovery step and filtered the most meaningful dependencies. Thereafter, we incrementally added tuples for each subsequent test batch, whose target attributes' classes are those predicted by the model. Specifically, the updated dataset was given in input to the discovery module to obtain updated sets of RFD_cs. Finally, we compared the original and the updated sets of dependencies by computing the metrics described in Section 5.

To evaluate the effectiveness of the proposed metrics, we compared, for each class, the correlation of their trend with the actual performance trend of the model in terms of *F1-Measure*. Notice that, for confusion matrix-based metrics, the *F1-Measure* is computed and compared with the model performances. The overall correlations on all classes are compared with the ones achieved by the baseline approaches. For the metrics that express the divergence between Σ_{y_i} and Σ'_{y_i} (including the baseline approaches) we expect a negative correlation, as the model performance should decrease when divergence increases. Conversely, for the confusion matrix-based metrics, we expect a positive correlation, as higher scores denote less difference among the compared sets of RFD_cs.

Dataset	Source	ID	Samp. Rows	Classes	Sel. Features	Drift
Followers	Kaggle	1	5000	2	11	Abrupt
		2	5000	2	11	Abrupt
		3	5000	2	11	Abrupt
		4	5000	2	11	Gradual
		5	5000	2	11	No Drift
		6	5000	2	11	Abrupt
		7	5000	2	11	Gradual
		8	5000	2	11	Abrupt
		9	5000	2	11	Gradual
Age	UCI ML	10	4500	2	9	🟢
		11	4500	2	9	🔺
		12	4500	2	9	🔶
Recruitment	Kaggle	13	2500	2	10	🟢
		14	2500	2	10	🔺
		15	2500	2	10	🔶
CovType	UCI ML	16	4000	2	12	🟢
		17	4000	2	12	🔺
		18	4000	2	12	🔶
Bankrupt	UCI ML	19	1000	2	9	Unknown
		20	1000	2	9	Unknown
Event Logs	[43]	21	365	3	20	Unknown
Cleveland	UCI ML	22	303	2	14	Unknown
KC2	Open ML	23	522	2	9	Unknown
Heart-Statlog	Open ML	24	270	2	10	Unknown

Table 6: Details of the datasets employed in the evaluation.

6.3 Experimental Results

In this section, we first discuss the experimental results observed on datasets with *Known Drift*, and then the ones obtained on datasets with *Unknown Drift*. To conclude the section, we provide an overall discussion of the outcomes of our study.

6.3.1 *Datasets with Known Drift.*

Figure 3 shows the distribution of correlations obtained by each metric in the configurations with *Known Drift* (IDs 1-18). These have been computed as the average of the correlations over all classes. The metrics are ordered on the x-axis according to the median correlation value, starting from the best towards the worst. Notice that, in order to show all metrics in a single plot, we considered the inverted correlation values for the confusion matrix-based metrics, since as expected they shown positive values.

In general, we can observe that most of the RFD-based metrics achieved stronger correlations than the baseline approaches. In fact, Hi_{mean}, He_{max}, and Hi_{max} present weaker correlations, while He_{mean} recorded slightly better results, outperforming three of the proposed metrics (i.e., D_9, CF_6, and CF_7). All other RFD-based divergences and confusion matrix-based metrics shown stronger correlations, with median values better than −0.9, confirming the validity of both approaches in quantifying drift in the data.

We found that the best metrics are two divergences (i.e., D_5 and D_7) and two confusion matrix-based metrics (i.e., CF_2 and CF_3), even though the latter exhibit a larger interquartile range, indicating more variability in the results. On the other hand, the best divergences show a narrower interquartile range, suggesting that they may be more consistent and reliable. This is reflected by analyzing the average correlation obtained by these metrics: D_5 and D_7 have an average correlation of −0.94 and −0.93, respectively, while CF_3 and CF_2 have an average correlation of 0.87 and 0.86, respectively. Concerning the baseline approaches, the best

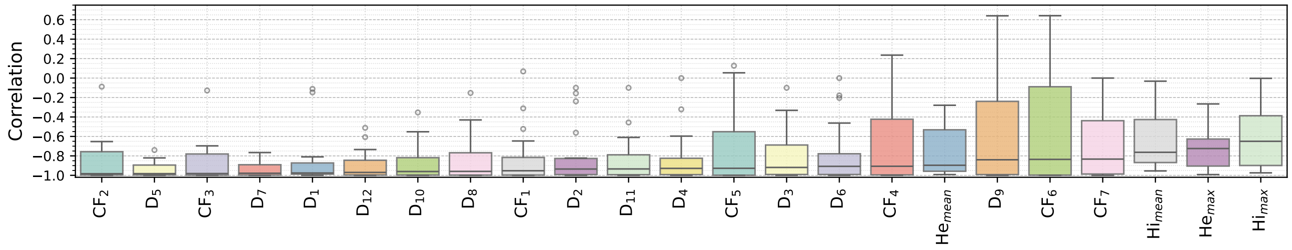

Figure 3: Correlations of metrics with the models' F1-Measure (the lower the better) on datasets with *Known Drift* (1-18).

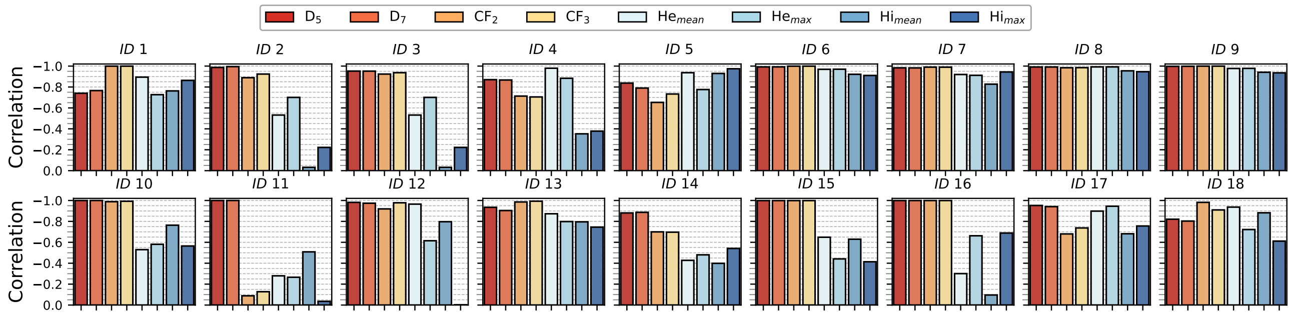

Figure 4: Correlations with the models' F1-Measure on datasets with *Known Drift*.

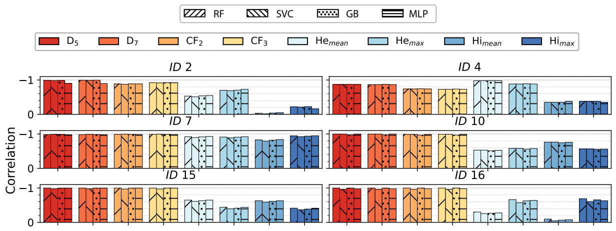

Figure 5: Correlations with different ML models.

metric was He_{mean}, with an average correlation of −0.75, while He_{max} achieved a slightly lower correlation (i.e., −0.73). Hi_{mean} and Hi_{max} performed significantly worse, with average correlations of −0.62 and −0.59, respectively. Thus, among the two aggregation strategies, averaging performed better than selecting the maximum distance (i.e., He_{mean} vs He_{max} and Hi_{mean} vs Hi_{max}). This could be due to the fact that the maximum aggregation strategy is likely more sensitive to outliers, overemphasizing single large shifts that may not impact the overall behavior of model's performance.

Figure 4 shows in detail the results of the individual experiments for the top 4 RFD-based metrics with respect to the baseline approaches. As described before, we divided this first phase of experiments in two parts to evaluate two different drift scenarios. The first row of results (IDs 1 to 9) relates to the experiments performed on the *Followers* dataset. In this scenario, we expected the baseline approaches to perform well, since the dataset contains changes of the statistical properties of individual attributes, which should represent the type of drift the baseline approaches better detect. This expectation was confirmed by the experimental results, in which the *Hellinger* distance performed reasonably well: He_{mean} achieved an average correlation of −0.86 and He_{max} a correlation of −0.85, with both metrics showing lower correlations only for experiments with IDs 2 and 3. Despite these good results, RFD-based metrics outperformed them. In fact, D_5 was the best metric, with an average correlation of −0.927, followed by D_7 (i.e., −0.926), CF_3 (i.e., 0.91), and CF_2 (i.e., 0.90). Instead, Hi_{mean} and Hi_{max}, recorded the worst results, with correlations of −0.71 and −0.64, respectively.

The second group of experiments (IDs 10-18) is shown in the second row of Figure 4. As discussed, we artificially introduced drift by shuffling the column values to alter multi-column relationships. This type of drift significantly affected the performance of baseline approaches, which were often unable to provide a correct drift estimation. He_{mean} achieved an average correlation of −0.65, while He_{max} and Hi_{mean} recorded a correlation of −0.61. Finally, Hi_{max} obtained the worst result, with an average correlation of −0.48. Thus, the trend of these distances was not aligned to the *F1-Measure* of the model. Instead, the RFD-based metrics showed the best results. In particular, D_5 and D_7 achieved an average correlation of −0.95 and −0.94, respectively; whereas CF_3 and CF_2 performed slightly worse, with correlations of 0.82 and 0.81, respectively. This decrease is mainly due to the negative results obtained in the experiment with ID 11, caused by a low correlation on one of the classes. Overall, we can conclude that although confusion matrix-based metrics are capable of obtaining significant results, they are less reliable than RFD-based divergences. Among the latter, D_5 and D_7 proved to be the most effective, consistently maintaining strong correlations across all experiments. To evaluate the generalizability of the results with respect to other ML models, we analyzed the correlations obtained using Random Forest (RF), Support Vector Classifier (SVC), Gradient Boosting Classifier (GB), and Multilayer Perceptron (MLP) on a sample of experiments. As shown in Figure 5, all metrics exhibit minimal variations. In fact, although the F1-Measure differs among models, the behavior of the latter is similar, i.e., they remain stable with normal data and degrade with drift-affected data.

6.3.2 *Datasets with Unknown drift.*

Figure 6 shows the correlations achieved by the top 4 RFD-based metrics on datasets with *Unknown Drift* (IDs 19-24), with respect to the ones obtained by the baseline approaches. As it can be seen, the RFD-based divergences D_5 and D_7 achieved the strongest correlations in almost all experiments, confirming themselves as the best metrics we proposed, achieving an average correlation of −0.946 and −0.948,

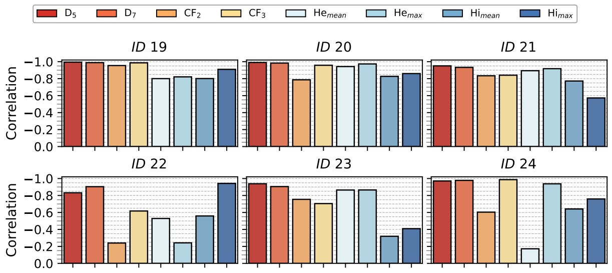

Figure 6: Correlations with the models' F1-Measure on datasets with *Unknown Drift*.

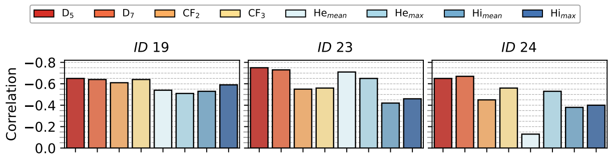

Figure 7: Correlations obtained by using smaller test batches.

respectively. Concerning confusion matrix-based metrics, we observed a similar behavior with respect to the previous datasets configurations. In fact, although they achieved good correlations in some configurations, their performances were less consistent, leading to a lower average outcome (i.e., 0.85 for CF_3 and 0.70 for CF_2). In particular, CF_2 has been affected by poor outcomes in experiments with IDs 22 and 24.

As for the baseline approaches, the results in this scenario diverged from those observed previously. In particular, although Hellinger was still the most effective baseline metric, in this scenario, the better aggregation strategy was selecting the highest value. In fact, He_{max} and Hi_{max} achieved an average correlation of −0.79 and −0.74, respectively, while He_{mean} and Hi_{mean} recorded a correlation of −0.70 and −0.65, respectively. The better results of He_{max} and Hi_{max} might be explained by a significant distribution change of an attribute that most influenced the model behavior in some of the tested configurations. However, also in this case, D_5, D_7, and CF_3 were more accurate in quantifying drifts with respect to all baseline approaches.

As further contribution, we investigated how correlations change when using smaller test batches. For a subset of experiments, we reorganized the test batches on the same data. Specifically, after using 25% of the data for training, we defined the first test batch as 5% of the dataset (i.e., from 25% to 30%), and the subsequent test batches as 10% each (e.g., from 30% to 40%, from 40% to 50%, and so froth). As shown in Figure 7, there has been a general decrease in correlations, due to the models exhibiting a more unstable trend. However, by comparing the correlation with the ones in Figure 6, the overall behavior of the metrics remains consistent, with D_5 and D_7 standing out as the most reliable metrics even in this scenario.

6.3.3 Discussion. The analyzed results shown that studying the evolution of RFDs over time can provide useful insights on whether the monitored model is providing accurate predictions or not. As for the divergence metrics, most of them showed a strong negative correlation with the performance of the model, indicating that their upward trend is very similar to the downward trend of the model. Among the proposed divergences, D_5 and D_7 were found to be the most robust and effective in quantifying drifts in the data on all the analyzed scenarios. As for confusion matrix-based metrics, most of them also exhibited a rather high correlation. This indicates that our reinterpretation of the confusion matrix to represent changes in RFDs is effective, and the trends of the *F1-Measure* computed on it are comparable to the ones of the monitored model. Among the proposed confusion matrix-based metrics, CF_3 and CF_2 achieved the best correlations with the models' performance in the analyzed scenarios. However, these metrics showed more instability than divergences, indicating that they need further refinements. One possible reason for this instability is that, unlike divergence metrics, confusion matrix-based metrics do not use weighting coefficients for the different ways in which RFDs can evolve. To summarize, we consider D_5 and D_7 to be the best metrics to use for quantifying concept drift, as they consistently demonstrated strong correlations with the performance of ML models. Although D_5 and D_7 differ slightly on how they evaluate specializations, their similar behavior suggests that they can be used interchangeably.

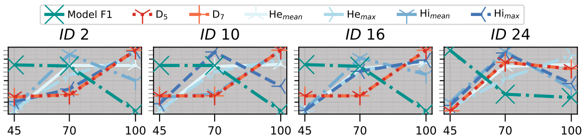

Figure 8: Trend analysis.

Experimental results also shown that most of the proposed metrics can provide more accurate estimates of drifts within data than other data distribution-based measures proposed in the literature. This is due to the fact that, although the latter demonstrated their capability in capturing changes in the single-column statistics of data, they did not account for changes between multi-attribute relationships. For this reason, we argue that RFD-based metrics better reflect model performance, minimizing the risk of false positives and false negatives when employed in a concept drift detection system. To support this claim, Figure 8 presents illustrative examples comparing the model's behavior with the divergences D_5 and D_7, as well as with the baseline metrics. The trends shown are averaged across all classes and have been z-normalized to enable a visual comparison. As it can be seen, even though in some experiments the performance of the model sharply dropped, the baseline metrics reported only slight changes or no changes; whereas in some other cases, they even tended to follow the model performance, leading to unexpected behaviors that make such distances unable to describe the model degradation in such a kind of scenarios. Examples of this behavior that can be recognized as false negative are observed in the third test batch of experiments with IDs 2, 10, 16, and 24. On the other hand, in some experiments baseline metrics reported distributional changes that had not affected model performances since the underlying relationships among the attributes possibly remained unchanged. Examples of this behavior that can be recognized as false positive are observed in the second test batch of experiments with IDs 2, 10, 16. In contrast to baseline metrics, RFD-based divergences were better able to describe the trend of the model, rising in case of performance degradation and remaining stable otherwise.

By concluding, we can state that in a real-world scenario where a model is deployed and it is not possible to obtain feedback on its predictions, RFD-based metrics can better provide insights into its performance trend over time.

7 CONCLUSION AND FUTURE WORKS

The transition of learning models from the training phase to the deployment in real-world contexts, requires to face the challenge of sustaining the models' effectiveness. In fact, the dynamic nature of data leads to possible shifts in them, causing a plausible decrease in the models' performance. An important aspect in these scenarios is to estimate the drift magnitude from data without the need of feedback on the model predictions. To this end, several approaches exploit measures to estimate the change in the data distribution. However, these approaches may not capture changes that occur in the relationships between attributes that may affect the performance of the model. In this paper, we investigated the potential of profiling metadata to assess the evolution of data over time. Specifically, we formalized the theoretical relationship between changes in terms of RFDs and performance trends of predictive models over time. Then, we introduced two categories of RFD-based metrics to measure the shift within data. In particular, they are based on sets of RFD$_c$s discovered from data collected in different time instants, as it happens with samples involved in training and deployment phases. Moreover, we introduced a suite of RFD-based divergences and a set of RFD confusion matrix-based metrics inspired by well-known ML measures. To evaluate the proposed metrics, we considered several datasets with both known and unknown drift, by also comparing them with other distribution-based measures. Results shown that the trend of RFD-based metrics is strongly correlated with the *F1-Measure* of the model, and that they provide more reliable insights than the compared baseline metrics, especially in more complex cases in which drift affects relationships between attributes. Drawing from empirical evidence, we established D_5 and D_7 as the most effective RFD-based metrics for accurately assessing drift.

In the future, we would like to investigate other types of profiling metadata to define a complete framework of drift metrics, which holistically consider statistics, patterns, and properties among data to monitor model performances. Furthermore, we plan to further refine confusion matrix-based metrics, aiming to improve their overall stability. While in this work we leveraged a static discovery algorithm, future works should employ incremental discovery strategies [6–8] to update RFD$_c$s over time without reconsidering already processed data to keep RFD$_c$s updated while reducing discovery times. This requires the defining incremental RFD discovery algorithms capable of also inferring similarity/distance thresholds.

REFERENCES

[1] Waqar Ali, Wenhong Tian, Salah Ud Din, Desire Iradukunda, and Abdullah Aman Khan. 2021. Classical and modern face recognition approaches: a complete review. Multimedia Tools and Applications 80, 3 (2021), 4825–4880.

[2] Robert Anderson, Yun Sing Koh, and Gillian Dobbie. 2018. Predicting concept drift in data streams using metadata clustering. In 2018 International Joint Conference on Neural Networks (IJCNN). IEEE, IEEE, Rio de Janeiro, Brazil, 1–8.

[3] Manuel Baena-Garcıa, José del Campo-Ávila, Raul Fidalgo, Albert Bifet, Ricard Gavalda, and Rafael Morales-Bueno. 2006. Early drift detection method. In Fourth international workshop on knowledge discovery from data streams, Vol. 6. Citeseer, ACM, Philadelphia Pennsylvania, 77–86.

[4] Alexei Baevski, Wei-Ning Hsu, Alexis Conneau, and Michael Auli. 2021. Unsupervised Speech Recognition. In Proceedings of the Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems NeurIPS 2021. , virtual, 27826–27839.

[5] Firas Bayram, Bestoun S Ahmed, and Andreas Kassler. 2022. From concept drift to model degradation: An overview on performance-aware drift detectors. Knowledge-Based Systems 245 (2022), 108632.

[6] Bernardo Breve, Loredana Caruccio, Stefano Cirillo, Vincenzo Deufemia, and Giuseppe Polese. 2023. IndiBits: Incremental Discovery of Relaxed Functional Dependencies using Bitwise Similarity. In Proceedings of the 2023 IEEE 39th International Conference on Data Engineering (ICDE). IEEE, Anaheim, CA, 1393–1405.

[7] Loredana Caruccio and Stefano Cirillo. 2020. Incremental discovery of imprecise functional dependencies. Journal of Data and Information Quality (JDIQ) 12, 4 (2020), 1–25.

[8] Loredana Caruccio, Stefano Cirillo, Vincenzo Deufemia, Giuseppe Polese, et al. 2019. Incremental Discovery of Functional Dependencies with a Bitvector Algorithm. In Proceedings of the 27th Italian Symposium on Advanced Database Systems. CEUR-WS.org, Castiglione della Pescaia (Grosseto), Italy, 12.

[9] Loredana Caruccio, Stefano Cirillo, Vincenzo Deufemia, Giuseppe Polese, and Roberto Stanzione. 2022. Data Analytics on Twitter for Evaluating Women Inclusion and Safety in Modern Society. In Proceedings of the 1st Italian Conference on Big Data and Data Science (ITADATA) 2022. CEUR-WS.org, Milan, Italy, 75–86.

[10] Loredana Caruccio, Vincenzo Deufemia, Felix Naumann, and Giuseppe Polese. 2020. Discovering relaxed functional dependencies based on multi-attribute dominance. IEEE Transactions on Knowledge and Data Engineering 33, 9 (2020), 3212–3228.

[11] Loredana Caruccio, Vincenzo Deufemia, and Giuseppe Polese. 2015. Relaxed functional dependencies—a survey of approaches. IEEE Transactions on Knowledge and Data Engineering 28, 1 (2015), 147–165.

[12] Loredana Caruccio, Vincenzo Deufemia, and Giuseppe Polese. 2020. Mining relaxed functional dependencies from data. Data Mining and Knowledge Discovery 34, 2 (2020), 443–477.

[13] Chun-Wei Chiang and Ming Yin. 2021. You'd better stop! Understanding human reliance on machine learning models under covariate shift. In Proceedings of the 13th ACM Web Science Conference 2021. ACM, United Kingdom, 120–129.

[14] Roberto Souto Maior de Barros, Danilo Rafael de Lima Cabral, Paulo Mauricio Gonçalves Jr., and Silas Garrido Teixeira de Carvalho Santos. 2017. RDDM: Reactive drift detection method. Expert Syst. Appl. 90 (2017), 344–355.

[15] Massimiliano De Benedetti, Fabio Leonardi, Fabrizio Messina, Corrado Santoro, and Athanasios Vasilakos. 2018. Anomaly detection and predictive maintenance for photovoltaic systems. Neurocomputing 310 (2018), 59–68.

[16] Gregory Ditzler and Robi Polikar. 2011. Hellinger distance based drift detection for nonstationary environments. In 2011 IEEE symposium on computational intelligence in dynamic and uncertain environments (CIDUE). IEEE, IEEE, Paris, France, 41–48.

[17] Pablo A. Estevez, Michel Tesmer, Claudio A. Perez, and Jacek M. Zurada. 2009. Normalized Mutual Information Feature Selection. IEEE Transactions on Neural Networks 20, 2 (2009), 189–201.

[18] Joao Gama, Pedro Medas, Gladys Castillo, and Pedro Rodrigues. 2004. Learning with drift detection. In Proceedings of the 17th Brazilian Symposium on Artificial Intelligence - Advances in Artificial Intelligence SBIA 2004. Proceedings 17. Springer, Sao Luis, Maranhao, Brazil, 286–295.

[19] Lukasz Golab, Howard J. Karloff, Flip Korn, Divesh Srivastava, and Bei Yu. 2008. On generating near-optimal tableaux for conditional functional dependencies. Proceeding of the VLDB Endow. 1, 1 (2008), 376–390.

[20] Igor Goldenberg and Geoffrey I Webb. 2019. Survey of distance measures for quantifying concept drift and shift in numeric data. Knowledge and Information Systems 60, 2 (2019), 591–615.

[21] Igor Goldenberg and Geoffrey I Webb. 2020. PCA-based drift and shift quantification framework for multidimensional data. Knowledge and Information Systems 62, 7 (2020), 2835–2854.

[22] Ömer Gözüaçık, Alican Büyükçakır, Hamed Bonab, and Fazli Can. 2019. Unsupervised concept drift detection with a discriminative classifier. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management. ACM, Beijing, China, 2365–2368.

[23] Ege Berkay Gulcan and Fazli Can. 2023. Unsupervised concept drift detection for multi-label data streams. Artificial Intelligence Review 56, 3 (2023), 2401–2434.

[24] Ernst Hellinger. 1909. Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen. Journal für die reine und angewandte Mathematik 1909, 136 (1909), 210–271.

[25] Mohammad Kazim Hooshmand and Doreswamy Hosahalli. 2022. Network anomaly detection using deep learning techniques. CAAI Transactions on Intelligence Technology 7, 2 (2022), 228–243.

[26] Dongxu Huang, Dejun Mu, Libin Yang, and Xiaoyan Cai. 2018. CoDetect: Financial fraud detection with anomaly feature detection. IEEE Access 6 (2018), 19161–19174.

[27] David Tse Jung Huang, Yun Sing Koh, Gillian Dobbie, and Russel Pears. 2014. Detecting volatility shift in data streams. In Prooceedings of the IEEE International Conference on Data Mining ICDM 2014. IEEE Computer Society, Shenzhen, China, 863–868.

[28] Lyudmyla F Kozachenko and Nikolai N Leonenko. 1987. Sample estimate of the entropy of a random vector. Problemy Peredachi Informatsii 23, 2 (1987), 9–16.

[29] Marie Le Guilly, Jean-Marc Petit, and Vasile-Marian Scuturici. 2020. Evaluating classification feasibility using functional dependencies. Transactions on Large-Scale Data-and Knowledge-Centered Systems XLIV: Special Issue on Data Management–Principles, Technologies, and Applications 44 (2020), 132–159.

[30] V Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. Proceedings of the Soviet physics doklady 1 (1966), 1–15.

[31] Chen Li, Jinha Park, Hahyeon Kim, and Dimitrios Chrysostomou. 2021. How can I help you? An Intelligent Virtual Assistant for Industrial Robots. In Companion of the 2021 ACM/IEEE International Conference on Human-Robot Interaction, HRI 2021. ACM, Boulder, CO, USA, 220–224.

[32] Shangdong Liu, Lili Lu, Yongpan Zhang, Tong Xin, Yimu Ji, and Ruchuan Wang. 2017. Research on concept drift detection for decision tree algorithm in the stream of big data. In Proceedings of the 8th International Symposium on Parallel Architecture, Algorithm and Programming PAAP 2017, Proceedings 8. Springer, Haikou, China, 237–246.

[33] Robert J. May, Holger R. Maier, Graeme C. Dandy, and T.M.K. Gayani Fernando. 2008. Non-linear variable selection for artificial neural networks using partial mutual information. Environmental Modelling & Software 23, 10 (2008), 1312–1326.

[34] Felix Naumann. 2013. Data profiling revisited. ACM SIGMOD Record 42, 4 (2013), 40–49.

[35] Kyosukef Nishida and Koichiro Yamauchi. 2007. Detecting concept drift using statistical testing. In Proceedings of the 10th International Conference on Discovery Science DS 2007. Springer, Sendai, Japan, 264–269.

[36] Ivens Portugal, Paulo S. C. Alencar, and Donald D. Cowan. 2018. The use of machine learning algorithms in recommender systems: A systematic review. Expert Systems with Applications 97 (2018), 205–227.

[37] Abdulhakim A Qahtan, Basma Alharbi, Suojin Wang, and Xiangliang Zhang. 2015. A pca-based change detection framework for multidimensional data streams: Change detection in multidimensional data streams. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, Sydney, Australia, 935–944.

[38] Edin Šabić, David Keeley, Bailey Henderson, and Sara Nannemann. 2021. Healthcare and anomaly detection: using machine learning to predict anomalies in heart rate data. AI & SOCIETY 36, 1 (2021), 149–158.

[39] Riyanarto Sarno, Fernandes Sinaga, and Kelly Rossa Sungkono. 2020. Anomaly detection in business processes using process mining and fuzzy association rule learning. Journal of Big Data 7, 1 (2020), 5.

[40] Jaime Céspedes Sisniega and Álvaro López García. 2024. Frouros: An open-source Python library for drift detection in machine learning systems. SoftwareX 26 (2024), 101733.

[41] Shaoxu Song and Lei Chen. 2013. Efficient discovery of similarity constraints for matching dependencies. Data & Knowledge Engineering 87 (2013), 146–166.

[42] Michael J Swain and Dana H Ballard. 1991. Color indexing. International journal of computer vision 7, 1 (1991), 11–32.

[43] Gabriel Marques Tavares and Sylvio Barbon Junior. 2021. Process mining encoding via meta-learning for an enhanced anomaly detection. In Proceedings of the European Conference on Advances in Databases and Information Systems ADBIS 2021. Springer, Tartu, Estonia, 157–168.

[44] Geoffrey I Webb, Roy Hyde, Hong Cao, Hai Long Nguyen, and Francois Petitjean. 2016. Characterizing concept drift. Data Mining and Knowledge Discovery 30, 4 (2016), 964–994.