POLITECNICO DI TORINO
Repository ISTITUZIONALE

Explaining Concept Drift via Neuro-Symbolic Rules

*Original*
Explaining Concept Drift via Neuro-Symbolic Rules / Basci, P., Greco, S., Manigrasso, F., Cerquitelli, T., Morra, L.. - ELETTRONICO. - 4132:(2025), pp. 61-72. (European Workshop on Trustworthy AI (TRUST-AI) Bologna (ITA) 25-26 ottobre 2025).

*Availability:*
This version is available at: 11583/3003603 since: 2025-12-16T12:58:15Z

*Publisher:*
CEUR

*Published*
DOI:

*Terms of use:*

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

*Publisher copyright*

(Article begins on next page)

02 November 2026

# Explaining Concept Drift via Neuro-Symbolic Rules

Pietro Basci[1], Salvatore Greco[1], Francesco Manigrasso[1], Tania Cerquitelli[1] and Lia Morra[1,*]

[1]Department of Control and Computer Engineering, Politecnico di Torino, Italy

**Abstract**

Concept drift in machine learning refers to changes in the underlying data distribution over time, which can lead to a degradation in the performance of predictive models. Although many methods have been proposed to detect and adapt to concept drift, effective methods to explain it in a human-understandable manner remain lacking. To address this, we propose the use of neuro-symbolic rules to explain the reason for drift. We applied recent rule extraction methods to convolutional neural networks (CNNs) to shed light on the model's internal behavior and promote interpretability of the outputs, while also proposing two novel automated approaches for semantic kernel labeling. We conducted preliminary experiments to assess the applicability and effectiveness of these rules in explaining concept drift, and the efficacy of the kernel labeling strategies. Under the optimality assumption, our method was able to extract rules that can facilitate the identification of the causes of drift, through either rule inspection or antecedents activation frequencies analysis. Moreover, the proposed strategies for kernel labeling offer a more reliable and scalable alternatives to the state-of-the-art solutions.

**Keywords**

Neuro-Symbolic AI, Explainable AI, Concept Drift, Data Drift, Explainable Concept Drift, Trustworthy AI

## 1. Introduction

Artificial Intelligence (AI)-based models are increasingly integrated into various aspects of our daily lives, serving as automated decision-making systems across many domains. These models are trained to predict the future based on historical data. However, since the world is dynamic and constantly evolving, the patterns and relationships learned during training can become outdated, resulting in a degradation of the model's performance over time. This phenomenon is known as *concept drift* [1].

To ensure production models remain reliable and robust even in changing environments, continuous monitoring for concept drift is essential for building trustworthy AI systems over time [2, 3, 4]. This encompasses *detecting* whether and when concept drift occurs, *explaining* the underlying changes that caused it, and *adapting* the model to maintain its performance [5, 6].

As we will discuss in Section 2, substantial research has been devoted to developing techniques for *detecting* concept drift—identifying whether and when drift occurs. Once a drift is detected, effective and human-readable explanations can enhance understanding and support adaptation [6]. However, comparatively less research has focused on *explaining* the underlying causes of drift.

Previous research on *drift explanation* has shown the potential of Explainable Artificial Intelligence (XAI) techniques for this purpose. Unlike traditional XAI methods that explain why a model makes specific predictions [7, 8, 9, 10], drift explanation approaches adapt XAI techniques to explain why drift has been detected [6, 11]. However, prior work has not explored the use of rule extraction from neural networks to produce drift explanations that are both interpretable and scalable, particularly for expert users. In this paper, we address this gap by proposing and evaluating the use of logic rules to explain concept drift. In doing so, we make two main contributions:

1. *Methodology* (Section 3): We propose the use of neuro-symbolic rules to explain the reasons for concept drift. By applying rule extraction methods to Convolutional Neural Networks (CNNs),

*TRUST-AI: The European Workshop on Trustworthy AI. Organized as part of the European Conference of Artificial Intelligence - ECAI 2025. October 2025, Bologna, Italy.*

*Corresponding author.

pietro.basci@polito.it (P. Basci); salvatore_greco@polito.it (S. Greco); francesco.manigrasso@polito.it (F. Manigrasso); tania.cerquitelli@polito.it (T. Cerquitelli); lia.morra@polito.it (L. Morra)

0009-0008-3335-6675 (P. Basci); 0000-0001-7239-9602 (S. Greco); 0000-0002-4151-8880 (F. Manigrasso); 0000-0002-9039-6226 (T. Cerquitelli); 0000-0003-2122-7178 (L. Morra)

© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

published 2025-12-16

CEUR-WS.org/Vol-4132/short54.pdf

we derive an interpretable set of rules that can be leveraged to formulate hypotheses regarding the causes of drift. We also introduce two scalable approaches for semantic kernel labeling which aim at mitigate the key limitations of the current systems.

2. *Experiment* (Section 4): We conducted a preliminary experiment using a deep learning classifier trained to predict the gender of individuals from input images. To simulate drift, we removed samples of *male individuals wearing earrings* from the training data and then introduced such images during inference. This drift led to a noticeable decrease in classifier performance on the drifted samples (accuracy drop = 0.09). Our method successfully extracts human-readable explanations in the form of logical rules, which explain the cause of the classifier's drift and its performance degradation (e.g., $if$ *Wearing Earrings* $\wedge$ $\neg$*Wearing Lipstick* $\wedge$ $\neg$*No Beard* $\rightarrow$ *Drift*).[1]

We conclude this work by discussing the limitations of our current approach and experiments, as well as outlining directions for future research (Section 5).

## 2. Background and Related Work

In this section, we first provide background on concept drift and review prior work in the field, with a particular focus on drift explanation (Section 2.1). We then discuss how to extract rules from neural networks, as we propose the use of these rules to explain concept drift (Section 2.2).

### 2.1. Concept Drift

Concept drift in machine learning refers to a change in the underlying data distribution over time, which can lead to a degradation in model performance [1]. Formally, concept drift can be defined as a change in the joint distribution over time, and occurs when: $P_t(X, y) \neq P_{t+w}(X, y)$, where $X$ are the feature vectors, $y$ is the target variable, $t$ is a given time point, and $w$ is the time window over which the distribution shift takes place [12]. Several sub-terms have been defined under concept drift, such as *real drift* (changes in $P(y/X)$), and *virtual* or *data drift* (changes in $P(X)$) [5]. Notice that, although concept drift is related to out-of-distribution detection (OOD) [13], there is a key distinction: OOD detection is primarily concerned with identifying individual samples that do not conform to the training feature distribution $P(X)$. In contrast, concept drift operates at the distribution level, often over a temporal window, and is not strictly related to changes in $P(X)$ only. In this paper, however, we adopt the general term *concept drift* as a collective term encompassing all such cases.

**Concept drift detection** Drift detectors aim to identify *whether* (and *when*) drift occurs, and quantify its severity [5]. Drift detection techniques can be categorized into two macro-categories: (i) *supervised* [1] and (ii) *unsupervised* [14, 15]. *Supervised* drift detection techniques assume the availability of ground-truth labels in the production data stream. They usually compute error rate-based measures or use ensemble models to monitor and detect performance decrease over time, such as an accuracy drop (e.g., [16, 17, 18, 19]). However, these techniques have limited applicability in real-world scenarios since ground-truth labels are usually unavailable. In contrast, *unsupervised* drift detection techniques do not require ground-truth labels to detect drifts. They usually apply statistical methods between two distributions [20, 21, 22, 23], or exploit model loss functions [24, 25, 26, 27], or train virtual classifiers [28, 29, 30] to detect drift. These techniques are generally more widely applicable, as newly processed data often lack ground-truth labels. However, they tend to be more resource-intensive, since they involve complex statistical tests and the training of additional models.

**Concept drift localization** Drift localization or segmentation techniques aim to identify the drift data points in the data space (*where*)–whether a given data point is affected by drift [6, 31]. This is usually obtained by quantifying the amount of drift in some regions of the data space or in each single data point by performing drift detection on a local scale, or by training virtual classifiers to distinguish between samples with or without drift.

[1]The code repository is available at: https://github.com/grecosalvatore/neurosymbolic-explainable-concept-drift

**Concept drift explanation** Some recent efforts have aimed to explain in human-readable terms the reasons for concept drift [6]. Some works provide a simpler form of drift visualization as feature-wise change intensity or change in correlation [32, 33, 34, 35, 36], or cause-effect analysis [35]. However, all these techniques provide more visualization than explanation of drift. Moreover, they usually struggle with high-dimensional data or non-semantic features, such as those in unstructured data like texts and images. In contrast, only a few works investigated the use of more advanced and readable Explainable Artificial Intelligence (XAI) [7, 8, 9, 10] techniques for explaining concept drift. [37, 38] proposes the use of SHAP [39] to characterise data drift. [40] proposes the identification of relevant prototype examples to explain the reasons for drift. Finally, [11] proposes training a classifier to differentiate between samples with and without drift, and then applying explainable AI (XAI) techniques—such as feature importance or counterfactual generation—to the classifier to explain the nature of the drift.

**Concept drift adaptation** Some techniques also propose methods to adapt to concept drift or make incremental changes to the model [41, 42, 43]. However, automatic drift adaptation remains particularly challenging [5, 6], especially in data streams with large volumes of unstructured data, such as images, that often lack ground-truth labels. In such cases, effective drift explanation and characterization can assist experts in annotating new samples to better adapt to drift.

In this paper, we focus on *drift explanation* only, assuming detection and localization have been completed. We target concept drift in CNN for image classification, aiming to provide effective, human-readable explanations by extracting rules that explain the reasons for changes.

### 2.2. Rule extraction from neural networks

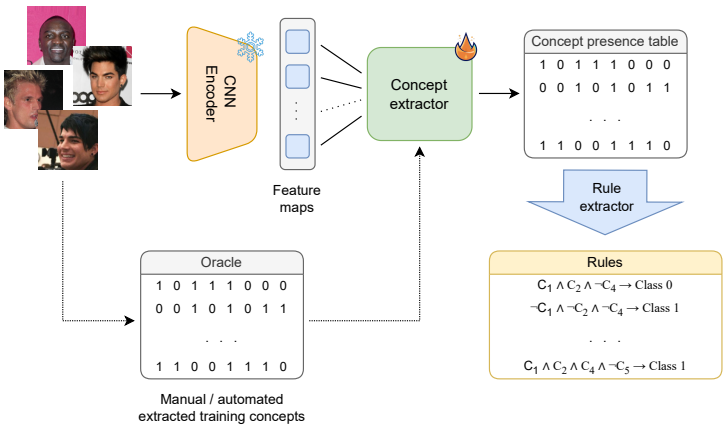
Deep convolutional neural networks (CNNs) have achieved remarkable performance in computer vision tasks, yet their internal decision-making remains largely opaque. To enhance transparency and accountability, research on explainable AI seeks methods that translate complex model behavior into human-understandable forms. One promising direction is rule extraction, which approximates network logic with a set of global, interpretable *if–then* rules. In contrast to local attribution methods that highlight individual/groups of pixels or neurons [44, 45, 46, 47], rule extraction provides a holistic description of the features and conditions driving model predictions. These rules serve as an interpretable surrogate for the original CNN, enabling practitioners to assess whether the model relies on semantically meaningful patterns or spurious correlations [48]. Early rule-extraction frameworks introduced taxonomies such as *pedagogical* (treating the network as a black box) and *decompositional* (leveraging internal structures), along with surrogate algorithms like the C4.5 decision tree [49]. More recent approaches extend these ideas to deep CNNs by training decision trees on the network's logits or feature activations to derive high-fidelity logical rules. For example, Padalkar et al. [50] present NeSyFOLD, a neurosymbolic architecture that replaces the final CNN layers with a response schedule of interpretable rules. Similarly, symbolic rule extraction has been applied to Vision Transformers (ViTs), where sparse attention maps and concept-level features yield human-readable rule sets [51].

## 3. Methodology

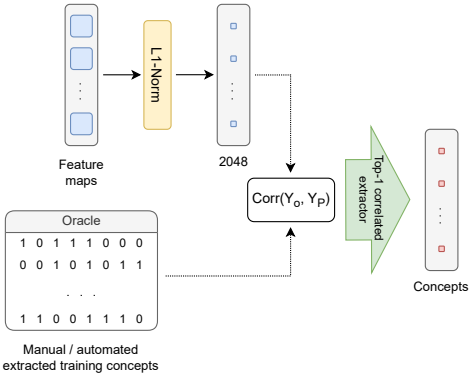
In this section, we describe the proposed concept drift explanation method, tailored to CNNs for image classification. We consider the original CNN model as-is and build upon the most recent advances in the field of neuro-symbolic rule extraction from neural networks [48, 50]. We assume that concept drift has already been detected and that the samples responsible for it have been identified—drift detection and localization are considered completed. Our objective is to extract neuro-symbolic rules that explain why such drifted samples differ from the normal data distribution, causing a drift.

### 3.1. Problem formulation

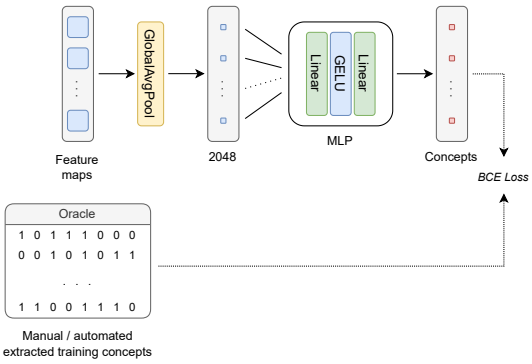
Consider an image $x \in \mathcal{X}$ with $\mathcal{X} \subset \mathbb{R}^{3\times H\times W}$ and a label $y \in \{0, 1\}$ for a binary classification problem. We assume $x$, which represents the main concept $y$, as a composition of multiple concepts




(a) Architecture for rule extraction.




(b) Correlation-based concept extractor.




(c) MLP-based concept extractor.

**Figure 1:** Our framework's architecture. **(a)**: The full architecture adopted for the rule extraction. Starting from the feature maps of the CNN encoder to be analyzed, the *Concept extractor* builds a *Concept presence table* by inferring the presence of concepts in the input images, which is used by the *Rule extractor* to derive the final rules. **(b)**: The architecture of the *Correlation-based Concept extractor*. It exploits the correlations between *Oracle* and *kernel norms* distributions to derive the *Concept presence table*. **(c)**: The architecture of the *MLP-based Concept extractor*. It employs a new trainable concept head to detect the presence of concepts from the feature maps.

$C : \{c_i\}_{i=1}^{n} \rightarrow y$. For instance, in a problem of gender classification, some concepts that may imply the gender *Male* are the presence of *beard*, absence of *makeup* or *lipstick* (see Section 4). From these concepts, it is possible to derive classification rules that are easily interpretable by humans, such as:

$$if\ Beard \wedge \neg Makeup \wedge \neg Lipstick \rightarrow Male$$

Similarly, the approach extends to other tasks, such as scene classification, where a given class (e.g., *camping*) is generally correlated to specific concept presence (e.g., *tent*, *lawn*, *woods*, *mountains*, etc.).

Pre-trained models, such as CNNs, may demonstrate unpredictable behavior when exposed to out-of-distribution inputs and data drift, leading to significant performance degradation. From this perspective, we focus on identifying the rule set $\{r_j\}_{j=1}^{m}$ that mimics the behavior of the model, and can highlight in a more interpretable way changes in predictive patterns caused by concept or data drift.

Each rule $r_j : \bigwedge_{k=1}^{n} a_k \rightarrow y_i$ is satisfied if all *antecedents*, expressed as positive $a_k$ or negative $\neg a_k$, evaluate true. In our setting, *antecedents* represent the presence (or absence) of a specific concept $c_i$ in the image, while the *consequent* $y_i$ is the final predicted class. Therefore, given a pre-trained model $M$, we aim to extract an interpretable rule-based model $\hat{M} : \{r_j\}_{j=1}^{m} \rightarrow \hat{M} \approx M$. This enables the identification of the satisfied rule that guides the model's final decision, and facilitates the detection of potentially failing antecedents–highlighting the corresponding parts of the network.

### 3.2. Model architecture

As stated above, the method considers the original CNN as-is and, starting from the feature maps from the last convolutional layer, it tries to infer the presence or absence of predefined concepts $c_i$ within the original image, producing a binarized *concept presence table*. Then, a rule extractor algorithm uses the sparse information of this table, pre-computed for the whole training dataset, to derive the rules that approximate the original model behavior. The complete architecture is shown in Figure 1a.

**Concept extraction and labeling** The extraction of concepts is performed using the feature maps from the last convolutional layer of the original CNN. We select this layer because it yields a more semantically rich representation. Typical approaches consist in binarizing the feature maps by computing the norms for all input samples and then applying a specific threshold for each kernel, defined as the mean [48] or the weighted sum of the mean and standard deviation [50] of the kernel norms distribution. A label is ultimately attributed to each kernel reflecting the most prominent concept among the images it responds to, by means of visual inspection [48] or by employing a semantic segmentation model [50] to detect the most relevant concept in the image. We propose two alternative approaches for concept extraction and labeling that leverage an external source of knowledge—an *Oracle*—defining the presence of predefined concepts in the training images. The *Oracle* can be manually annotated by humans or synthetically generated through pre-trained task-specific models for object detection, semantic segmentation, Visual Question Answering (VQA), or Vision Language Models (VLMs).

The first method (Figure 1b) is inspired by existing approaches, but introduces two major variations in the choice of thresholds for binarization and in the kernel labeling procedure. Specifically, starting from the feature maps $\{f_j\}_{j=1}^{k}$, we extract the kernel activations using the $L_1$-norm $\{a_j\}_{j=1}^{k} = \{||f_j||_1\}_{j=1}^{k}$, where $k$ is the number of kernels. We then compute the point-biserial [52] correlation coefficient $r_{pb} = Corr(Y_O, Y_P)$ between the dichotomous variable $Y_O$ from the *Oracle* and the continuous variable $Y_P$ representing the kernel norms, pre-computed for the whole training set. We repeat the process for all combinations and then select the pairs of concepts-kernels that exhibit the highest correlation:

$$\{(c, a)_i\}_{i=1}^{d} = \{\arg\max_{c_i, a_j}(r_{pb})\}_{i=1,j=1}^{d,k}$$

where $d$ is the number of concepts in the *Oracle*. In this way, we find the most representative kernel for each concept while, at the same time, achieving the kernel labeling. Binarization was finally achieved using per-concept percentiles computed on the *Oracle* to take into account the concept imbalance in the training data, and then the thresholds $\{\theta_i\}_{i=1}^{d}$ were selected to segment the bimodal norm distributions in a way that preserves the original cardinalities of each split.

The second method (Figure 1c) employs an MLP to infer the presence of concepts from feature maps, in a pure data-driven approach. Specifically, the results of the global average pooling were fed to the new concept head, which outputs binary values, one for each of the predefined concepts. The network was trained to match the *Oracle* using a BinaryCrossEntropy objective. Although the correlation-based extractor appears to be a viable solution, the second one is generally more accurate.

**Rule extraction** Once the binary *concept presence table* is available, the set of predicted concepts $C$ that define the *antecedents*, can be used to derive the prediction rules. Similar to [48], we employ a tree-based extraction algorithm, near to C4.5 [49], to derive rules that outline the conditions driving the model decisions. Specifically, to obtain an approximation $\hat{M}$ of the original model $M$, the algorithm was fitted to follow its predictions $\hat{Y} = M(X)$. Although, as proposed in [48], the rule extraction could be expanded to multiple layers, we limited the analysis to the final classifier of the CNN, which only exploits the high-level features of the final convolutional layer, as in [50]. In addition to the original model, we explored generating drift explanations by extracting rules that explain the behavior of a drift location model, which differentiates between drifted and non-drifted samples (see Section 4.2).

**Inference** Inference is straightforward, as it consists of extracting the feature maps using the original CNN, identifying the concepts using the *Concept extractor* module, and finding the *satisfied rule* based on the combination of activated *antecedents*.

**Table 1**
Accuracy drop caused by drift.

| | Acc ↑ | |
|---|---|---|
| **Drift simulation** | Data stream (0% drift) | Data stream (100% drift) |
| `Male, Wearing Earrings` | .98 | .89 |

```
¬Wearing_Lipstick & ¬No_Beard & ¬Wearing_Earrings & ¬Blurry & ¬Big_Nose → Male (conf=1.00)
¬Wearing_Lipstick & ¬No_Beard & ¬Wearing_Earrings & ¬Blurry & Big_Nose → Male (conf=1.00)
¬Wearing_Lipstick & ¬No_Beard & ¬Wearing_Earrings & Blurry & ¬Big_Nose → Male (conf=0.98)
¬Wearing_Lipstick & ¬No_Beard & ¬Wearing_Earrings & Blurry & Big_Nose → Male (conf=1.00)
¬Wearing_Lipstick & ¬No_Beard & Wearing_Earrings → ¬Male (conf=1.00)
¬Wearing_Lipstick & No_Beard & ¬Wearing_Earrings & ¬Wearing_Necktie & ¬Big_Nose → Male (conf=0.69)
¬Wearing_Lipstick & No_Beard & ¬Wearing_Earrings & ¬Wearing_Necktie & Big_Nose → Male (conf=0.91)
¬Wearing_Lipstick & No_Beard & ¬Wearing_Earrings & Wearing_Necktie & ¬Big_Nose → Male (conf=1.00)
¬Wearing_Lipstick & No_Beard & ¬Wearing_Earrings & Wearing_Necktie & Big_Nose → Male (conf=1.00)
¬Wearing_Lipstick & No_Beard & Wearing_Earrings → ¬Male (conf=1.00)
Wearing_Lipstick & ¬Heavy_Makeup & ¬Wearing_Earrings & ¬Bushy_Eyebrows & ¬Bangs → ¬Male (conf=0.98)
Wearing_Lipstick & ¬Heavy_Makeup & ¬Wearing_Earrings & ¬Bushy_Eyebrows & Bangs → ¬Male (conf=0.93)
Wearing_Lipstick & ¬Heavy_Makeup & ¬Wearing_Earrings & Bushy_Eyebrows & ¬Bangs → ¬Male (conf=0.87)
Wearing_Lipstick & ¬Heavy_Makeup & ¬Wearing_Earrings & Bushy_Eyebrows & Bangs → ¬Male (conf=0.58)
Wearing_Lipstick & ¬Heavy_Makeup & Wearing_Earrings → ¬Male (conf=1.00)
Wearing_Lipstick & Heavy_Makeup & ¬Wearing_Earrings & ¬Black_Hair & ¬Arched_Eyebrows → ¬Male (conf=1.00)
Wearing_Lipstick & Heavy_Makeup & ¬Wearing_Earrings & ¬Black_Hair & Arched_Eyebrows → ¬Male (conf=1.00)
Wearing_Lipstick & Heavy_Makeup & ¬Wearing_Earrings & Black_Hair & ¬Wearing_Necktie → ¬Male (conf=0.99)
Wearing_Lipstick & Heavy_Makeup & ¬Wearing_Earrings & Black_Hair & Wearing_Necktie → Male (conf=0.82)
Wearing_Lipstick & Heavy_Makeup & Wearing_Earrings → ¬Male (conf=1.00)
```

Listing 1: Rule extracted to approximate the gender classifier (optimal case). For visualization purposes, the maximum depth of the decision tree was set to 5, ensuring a fidelity of about 90%.

```
¬Wearing_Earrings → ¬Drift (conf=1.00)
Wearing_Earrings & ¬Wearing_Lipstick & ¬No_Beard → Drift (conf=1.00)
Wearing_Earrings & ¬Wearing_Lipstick & No_Beard & ¬Brown_Hair & ¬Bangs → Drift (conf=0.97)
Wearing_Earrings & ¬Wearing_Lipstick & No_Beard & ¬Brown_Hair & Bangs → Drift (conf=0.81)
Wearing_Earrings & ¬Wearing_Lipstick & No_Beard & Brown_Hair & ¬Smiling → Drift (conf=0.91)
Wearing_Earrings & ¬Wearing_Lipstick & No_Beard & Brown_Hair & Smiling → Drift (conf=0.56)
Wearing_Earrings & Wearing_Lipstick & ¬Bushy_Eyebrows & ¬Pointy_Nose & ¬Attractive → ¬Drift (conf=0.78)
Wearing_Earrings & Wearing_Lipstick & ¬Bushy_Eyebrows & ¬Pointy_Nose & Attractive → ¬Drift (conf=0.96)
Wearing_Earrings & Wearing_Lipstick & ¬Bushy_Eyebrows & Pointy_Nose → ¬Drift (conf=1.00)
Wearing_Earrings & Wearing_Lipstick & Bushy_Eyebrows & ¬Pointy_Nose & ¬Wavy_Hair → Drift (conf=0.80)
Wearing_Earrings & Wearing_Lipstick & Bushy_Eyebrows & ¬Pointy_Nose & Wavy_Hair → ¬Drift (conf=1.00)
Wearing_Earrings & Wearing_Lipstick & Bushy_Eyebrows & Pointy_Nose & ¬Oval_Face → ¬Drift (conf=0.50)
Wearing_Earrings & Wearing_Lipstick & Bushy_Eyebrows & Pointy_Nose & Oval_Face → ¬Drift (conf=1.00)
```

Listing 2: Rule extracted to approximate the drift classifier (optimal case). For visualization purposes, the maximum depth of the decision tree was set to 5, ensuring a fidelity of about 97%.

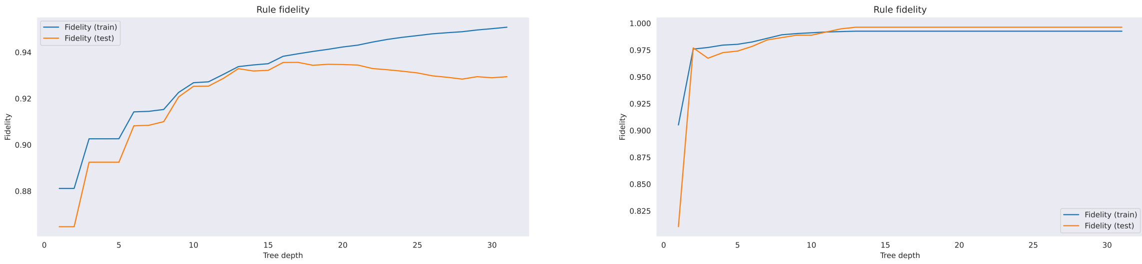
# 4. Experiments

In this section, we first introduce the experimental setting (Section 4.1). Then, we discuss the drift explanation results obtained by rules extraction (Section 4.2).

## 4.1. Experimental settings

**Dataset** We conduct our experiments using the CelebA dataset [53], which consists of about 202k images and provides human-annotated labels for 40 facial attributes, that can serve as the concept *Oracle* in the proposed framework.

**Task definition and drift simulation** We define the main task as a gender classification problem by choosing the attribute *Male* as the target. We simulate a drift by isolating all instances from the class *Male* that *wear earrings* and then injecting them into the data stream. In particular, the dataset was divided into 4 splits preserving the original distribution: *Historical train* of about 156k samples (used for training the CNN), *Historical test* of about 19k samples (used for testing the CNN), *Datastream* of




(a) Gender classification.

(b) Drift classification.

**Figure 2:** Rule fidelity (optimal case) as a function of the tree depth for the two considered settings.

**Table 2**
Antecedents activation frequencies on samples in the first experimental setting. The highest value highlights the most likely causes of the drift, which induces erroneous behavior in the model.

| | | Antecedents | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Drift simulation** | Drift | 5 o Clock Shadow | Arched Eyebrows | Attractive | Bags Under Eyes | Bald | Bangs | Big Lips | Big Nose | Black Hair | Blond Hair | Blurry | Brown Hair | Bushy Eyebrows | Chubby | Double Chin | Eyeglasses | Goatee | Gray Hair | Heavy Makeup | High Cheekbones | Mouth Slightly Open | Mustache | Narrow Eyes | No Beard | Oval Face | Pale Skin | Pointy Nose | Receding Hairline | Rosy Cheeks | Sideburns | Smiling | Straight Hair | Wavy Hair | Wearing Earrings | Wearing Hat | Wearing Lipstick | Wearing Necklace | Wearing Necktie | Young |
| `Male, W Earrings` | 0% | .00 | .00 | .06 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .01 | .01 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .02 | .00 | .00 | .07 | .00 | .00 | .00 | .01 | .00 | .00 | .19 | .00 | *.17* | .00 | .00 | .00 |
| | 100% | .00 | .00 | .00 | .00 | .00 | .02 | .00 | .00 | .00 | .00 | .00 | .03 | .01 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .38 | .00 | .00 | .00 | .00 | .00 | .00 | .01 | .00 | .00 | **1.0** | .00 | .03 | .00 | .00 | .00 |
| `Male, W Earrings` | 0% | .00 | .00 | .04 | .00 | .00 | .01 | .00 | .00 | .00 | .00 | .00 | .01 | .16 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .10 | .00 | .00 | .00 | .00 | .00 | .00 | **.81** | .00 | .02 | .00 | .00 | .00 |
| | 100% | .00 | .00 | .01 | .00 | .00 | .33 | .00 | .00 | .00 | .00 | .00 | .35 | .01 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | *.59* | .00 | .00 | .02 | .00 | .00 | .00 | .02 | .00 | .01 | .00 | .00 | .97 | .00 | .00 | .00 |
| | Drift | ¬5 o Clock Shadow | ¬Arched Eyebrows | ¬Attractive | ¬Bags Under Eyes | ¬Bald | ¬Bangs | ¬Big Lips | ¬Big Nose | ¬Black Hair | ¬Blond Hair | ¬Blurry | ¬Brown Hair | ¬Bushy Eyebrows | ¬Chubby | ¬Double Chin | ¬Eyeglasses | ¬Goatee | ¬Gray Hair | ¬Heavy Makeup | ¬High Cheekbones | ¬Mouth Slightly Open | ¬Mustache | ¬Narrow Eyes | ¬No Beard | ¬Oval Face | ¬Pale Skin | ¬Pointy Nose | ¬Receding Hairline | ¬Rosy Cheeks | ¬Sideburns | ¬Smiling | ¬Straight Hair | ¬Wavy Hair | ¬Wearing Earrings | ¬Wearing Hat | ¬Wearing Lipstick | ¬Wearing Necklace | ¬Wearing Necktie | ¬Young |

**Table 3**
Antecedents activation frequencies in the second experimental setting. The highest value highlights the most likely causes of the drift, which induces erroneous behavior in the model.

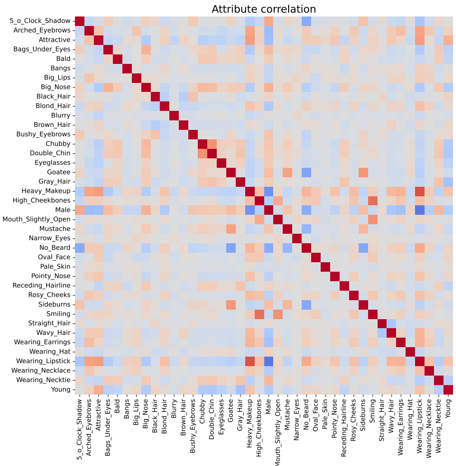
| | | Antecedents | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Drift simulation** | Drift | 5 o Clock Shadow | Arched Eyebrows | Attractive | Bags Under Eyes | Bald | Bangs | Big Lips | Big Nose | Black Hair | Blond Hair | Blurry | Brown Hair | Bushy Eyebrows | Chubby | Double Chin | Eyeglasses | Goatee | Gray Hair | Heavy Makeup | High Cheekbones | Mouth Slightly Open | Mustache | Narrow Eyes | No Beard | Oval Face | Pale Skin | Pointy Nose | Receding Hairline | Rosy Cheeks | Sideburns | Smiling | Straight Hair | Wavy Hair | Wearing Earrings | Wearing Hat | Wearing Lipstick | Wearing Necklace | Wearing Necktie | Young |
| `Male, W Earrings` | 0% | .00 | .09 | .00 | .00 | .00 | .01 | .00 | .18 | .03 | .00 | .01 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .39 | .00 | .00 | .00 | .00 | .38 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .19 | .00 | *.46* | .00 | .05 | .00 |
| | 100% | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .02 | .00 | .00 | .00 | .00 | .38 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | **1.0** | .00 | .03 | .00 | .00 | .00 |
| `Male, W Earrings` | 0% | .00 | .12 | .00 | .00 | .00 | .04 | .00 | .33 | .21 | .00 | .15 | .00 | .05 | .00 | .00 | .00 | .00 | .00 | .07 | .00 | .00 | .00 | .00 | .16 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | **.81** | .00 | .54 | .33 | .00 | .00 |
| | 100% | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .01 | .00 | .00 | .00 | .00 | *.59* | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .00 | .97 | .00 | .00 | .00 |
| | Drift | ¬5 o Clock Shadow | ¬Arched Eyebrows | ¬Attractive | ¬Bags Under Eyes | ¬Bald | ¬Bangs | ¬Big Lips | ¬Big Nose | ¬Black Hair | ¬Blond Hair | ¬Blurry | ¬Brown Hair | ¬Bushy Eyebrows | ¬Chubby | ¬Double Chin | ¬Eyeglasses | ¬Goatee | ¬Gray Hair | ¬Heavy Makeup | ¬High Cheekbones | ¬Mouth Slightly Open | ¬Mustache | ¬Narrow Eyes | ¬No Beard | ¬Oval Face | ¬Pale Skin | ¬Pointy Nose | ¬Receding Hairline | ¬Rosy Cheeks | ¬Sideburns | ¬Smiling | ¬Straight Hair | ¬Wavy Hair | ¬Wearing Earrings | ¬Wearing Hat | ¬Wearing Lipstick | ¬Wearing Necklace | ¬Wearing Necktie | ¬Young |

about 19k samples (used to simulate non-drifted data), *Datastream drift* of about 1.4k samples (used to simulate drifted data). Table 1 highlights the model performance degradation caused by drift in this configuration, resulting in a 0.09 drop in accuracy. Similarly, other drifts can be simulated by isolating instances with a specific concept or combination of concepts.

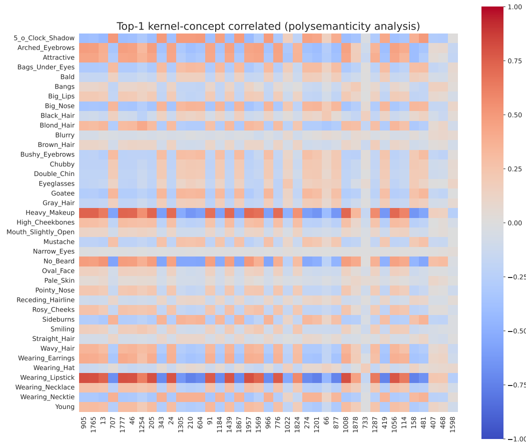
**CNN model and training details** We use the ResNet-50 architecture for our experiments. We train the model from scratch on the *Historical data* split for the gender classification task, using BCE loss for 5 epochs. We use the SGD with momentum optimizer with learning rate 1e-2, step decay of 0.1 after 3 epochs, and a batch size of 128. The MLP-based concept extractor is instead trained on top of the frozen CNN encoder, on the same split for the attribute recognition task, using BCE loss for 15 epochs. We use the AdamW optimizer, with learning rate 1e-3, step decay of 0.1 every 5 epochs, and a batch size of 256.

## 4.2. Rule extraction and drift explanation

To understand the expressiveness of the rules and their utility for the problem of drift explanation, we evaluate the rule extraction under the assumption of a perfect Concept extractor (i.e., *Concept presence*




(a) Attribute correlation.




(b) Top-1 kernel-concept correlated.

**Figure 3:** Correlation analysis on attributes and kernels. **(a):** Attribute correlation patterns in the CelebA dataset [53]. **(b):** Correlation patterns between top-1 binarized kernels and attributes.

**Table 4**

Concepts extraction accuracies. Results are calculated against historical data (1), data stream with 0% of drift samples (2), and data stream with 100% of drift samples (3).

| | Per-concept Acc ↑ | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Method** | 5 o Clock Shadow | Arched Eyebrows | Attractive | Bags Under Eyes | Bald | Bangs | Big Lips | Big Nose | Black Hair | Blond Hair | Blurry | Brown Hair | Bushy Eyebrows | Chubby | Double Chin | Eyeglasses | Goatee | Gray Hair | Heavy Makeup | High Cheekbones | Mouth Slightly Open |
| $\mathtt{Corr}_1$ | .88 | .75 | .71 | .75 | .95 | .77 | .68 | .75 | .69 | .84 | .90 | .72 | .82 | .90 | .92 | .90 | .91 | .92 | .84 | .65 | .57 |
| $\mathtt{Corr}_2$ | .88 | .75 | .70 | .75 | .96 | .78 | .70 | .74 | .70 | .83 | .91 | .70 | .82 | .90 | .92 | .89 | .91 | .91 | .84 | .64 | .56 |
| $\mathtt{Corr}_3$ | .79 | .79 | .84 | .61 | .89 | .91 | .43 | .51 | .49 | .98 | .91 | .95 | .74 | .69 | .83 | .69 | .69 | .93 | .97 | .56 | .46 |
| $\mathtt{MLP}_1$ | .91 | .79 | .77 | .81 | .98 | .90 | .77 | .81 | .86 | .93 | .95 | .82 | .88 | .95 | .96 | .97 | .95 | .96 | .88 | .82 | .76 |
| $\mathtt{MLP}_2$ | .91 | .79 | .76 | .81 | .98 | .90 | .86 | .79 | .87 | .92 | .95 | .80 | .89 | .95 | .95 | .97 | .95 | .96 | .88 | .82 | .76 |
| $\mathtt{MLP}_3$ | .83 | .79 | .86 | .73 | .91 | .96 | .43 | .60 | .67 | .98 | .98 | .96 | .80 | .73 | .87 | .90 | .76 | .98 | .97 | .75 | .72 |

| | Per-concept Acc ↑ | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Method** | Mustache | Narrow Eyes | No Beard | Oval Face | Pale Skin | Pointy Nose | Receding Hairline | Rosy Cheeks | Sideburns | Smiling | Straight Hair | Wavy Hair | Wearing Earrings | Wearing Hat | Wearing Lipstick | Wearing Necklace | Wearing Necktie | Young | **Acc ↑** | **AUC ↑** |
| $\mathtt{Corr}_1$ | .94 | .80 | .84 | .65 | .92 | .67 | .87 | .89 | .92 | .60 | .69 | .69 | .76 | .92 | .88 | .81 | .89 | .75 | .80 | - |
| $\mathtt{Corr}_2$ | .93 | .83 | .85 | .65 | .92 | .67 | .87 | .88 | .91 | .60 | .71 | .70 | .76 | .93 | .88 | .80 | .88 | .73 | .80 | - |
| $\mathtt{Corr}_3$ | .75 | .67 | .52 | .66 | .91 | .94 | .67 | 1.0 | .89 | .56 | .64 | .93 | .00 | .74 | .96 | .87 | .87 | .60 | .74 | - |
| $\mathtt{MLP}_1$ | .97 | .88 | .91 | .74 | .96 | .73 | .93 | .94 | .95 | .85 | .79 | .75 | .83 | .96 | .91 | .88 | .93 | .84 | .88 | .87 |
| $\mathtt{MLP}_2$ | .96 | .92 | .91 | .74 | .96 | .73 | .93 | .93 | .94 | .85 | .80 | .76 | .83 | .96 | .90 | .88 | .93 | .81 | .88 | .87 |
| $\mathtt{MLP}_3$ | .76 | .86 | .70 | .66 | .99 | .95 | .76 | 1.0 | .88 | .80 | .90 | .93 | .00 | .85 | .96 | .87 | .91 | .70 | .81 | .76 |

*table* $=$ *Oracle*) to avoid antecedent labeling mistakes that may compromise the rule assessment. We then evaluate the performance of the two proposed methods for Concept extraction and labeling to get an estimate of the gap with respect to the optimal case.

Starting from the optimal *Concept presence table* we extract the rules considering two different settings: (i) we fit the tree-based extraction algorithm to match the original model predictions and obtain an approximation of its behavior; (ii) we proceed with the assumption that drift has been successfully detected and localized (e.g., through established drift detectors [1, 5, 6]) and fit the extraction algorithm to predict whether individual samples are affected by drift. In both cases, we assume a perfect classification.

We aim to provide two complementary perspectives on model behavior: (i) **Human interpretability**: We extract interpretable rules that enable direct inspection to identify problematic associations that degrade model performance on concept drift and out-of-distribution samples (e.g., the model incorrectly relies on the presence of *Earrings* to predict that gender is *not Male*), and identify the specific responsible network component (kernel); and (ii) **Automated diagnosis**: We analyze antecedent activation frequencies to systematically identify faulty antecedents driving incorrect predictions, under the assumption that the most frequently activated antecedent is most likely responsible for the observed drift.

Listings 1 and 2 show the rules extracted in the two considered settings under the optimality assumption. Their interpretability facilitates an understanding of the antecedents primarily influencing the decision process. For instance, in Listing 1, two of the most critical rules are Rule 5 and 10, which state that the presence of *earrings*, regardless of the presence of *beard*, is sufficient to predict that the gender is *not Male*. In Listing 2, instead, one of the most informative is Rule 1 which states that in absence of *earrings* no drift is detected while, in the other case, the presence of drift depends on other factors that exhibit correlation with the gender class. For instance, Rule 2 states that the presence of *earrings* and *beard*, along with the absence of *lipstick*, is a sufficient condition to identify a drifted subset.

Figure 2 illustrates how the extracted rules approximate the original models for varying tree depths in the evaluated settings under the optimal assumption. In both cases, the rule fidelity remains high, ensuring a close representation of the original model's behavior with minimal degradation. Tables 2 and 3 present the activation frequencies of antecedents on data streams containing 0% and 100% drift samples, in both experimental settings. The results indicate that the satisfied rule leading to the final prediction consistently includes the antecedent *Wearing Earring*—the concept used to simulate the drift.

Finally, we evaluate the effectiveness of the Concept Extractor with respect to both concept sensitivity and concept labeling. Figure 3 shows the correlation in the CelebA dataset (a) and the correlation levels between *Oracle* concepts and kernels (top-1 pairs) in the correlation-based solution (b). The heatmap in Figure 3b highlights the problem of the polysemantic nature of kernel activations, which, in general, are sensitive to different correlated concepts. We attribute this problem to the lower performance of the correlation-based solution with respect to the MLP-based solution, which can leverage all the kernels and learn to weight them effectively to further reduce the error. Table 4, instead, shows the accuracies of the two proposed methods. Although generally both methods appear to be more reliable on larger concepts, results highlight the difficulties on very small concepts represented with few pixels—such as *male earrings*, which are typically much smaller than their female counterparts. Such fine details can easily be lost during convolutional operations.

## 5. Discussion

Our preliminary results show the potential relationship between model interpretability and concept drift explanation in the context of trustworthy AI systems. Through neuro-symbolic rule extraction, the proposed approach clarifies the model decision pathway and helps pinpoint the origin of mispredictions, with traceability down to the responsible network component (kernel). Under the optimality assumption, the extracted rules appear highly informative, making it easier to formulate hypotheses about the possible causes of drift. However, in a real scenario, the interpretability of those rules may be affected by the concept extraction and naming procedure due to the inherent flaws still present in these systems. We proposed two effective automated approaches to address the problem of concept labeling, which is one of the most challenging and still open problems of rule extraction systems, and provide an estimation measure of their reliability. By considering the annotations (*Oracle*) previously extracted on the whole training set, they not only allow to avoid the search for the top-k images to which each kernel reacts most and the visual inspection phase for the labeling process, but they also enable to enlarge the set of samples considered for the kernel-concept association, resulting in a more reliable and scalable solution.

**Limitations and future works** The efficacy of the method is inherently constrained by the a priori selection of concepts, which may obscure the true cause of the drift. Those systems can be susceptible to errors in the kernel labeling procedure, which can produce erroneously named *antecedents* that make the rule unreasonable, although the final classification is correct. The issue becomes more pronounced when small concepts are present in the images, as kernels in the final layers may lack sufficient sensitivity to detect them. Moreover, since the encoder was originally trained for a different objective, individual feature maps may not necessarily be sensitive to a single concept, a challenge only partially mitigated by the proposed naming strategies. In future work, we plan to (1) enhance the robustness of the rule extraction process for drift explanation, including the exploration of automatic concept labeling methods, (2) extend the methodology to additional data modalities (e.g., text, audio), and (3) conduct a more comprehensive experimental evaluation across diverse drift scenarios, models, and data types.

## Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT and Grammarly in order to: Grammar and spelling check.

## References


[1] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, G. Zhang, Learning under concept drift: A review, IEEE Transactions on Knowledge and Data Engineering 31 (2019) 2346–2363. doi:10.1109/TKDE.2018.2876857.

[2] F. Bayram, B. S. Ahmed, Towards trustworthy machine learning in production: An overview of the robustness in mlops approach, ACM Comput. Surv. 57 (2025). URL: https://doi.org/10.1145/3708497. doi:10.1145/3708497.

[3] B. Li, P. Qi, B. Liu, S. Di, J. Liu, J. Pei, J. Yi, B. Zhou, Trustworthy ai: From principles to practices, ACM Comput. Surv. 55 (2023). URL: https://doi.org/10.1145/3555803. doi:10.1145/3555803.

[4] J. Klaise, A. V. Looveren, C. Cox, G. Vacanti, A. Coca, Monitoring and explainability of models in production, 2020. URL: https://arxiv.org/abs/2007.06299. arXiv:2007.06299.

[5] F. Hinder, V. Vaquet, B. Hammer, One or two things we know about concept drift—a survey on monitoring in evolving environments. part a: detecting concept drift, Frontiers in Artificial Intelligence Volume 7 - 2024 (2024). URL: https://www.frontiersin.org/journals/artificial-intelligence/articles/10.3389/frai.2024.1330257. doi:10.3389/frai.2024.1330257.

[6] F. Hinder, V. Vaquet, B. Hammer, One or two things we know about concept drift—a survey on monitoring in evolving environments. part b: locating and explaining concept drift, Frontiers in Artificial Intelligence Volume 7 - 2024 (2024). URL: https://www.frontiersin.org/journals/artificial-intelligence/articles/10.3389/frai.2024.1330258. doi:10.3389/frai.2024.1330258.

[7] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A survey of methods for explaining black box models, ACM Comput. Surv. 51 (2018). URL: https://doi.org/10.1145/3236009. doi:10.1145/3236009.

[8] F. Xu, H. Uszkoreit, Y. Du, W. Fan, D. Zhao, J. Zhu, Explainable ai: A brief survey on history, research areas, approaches and challenges, in: Natural Language Processing and Chinese Computing, Springer International Publishing, Cham, 2019, pp. 563–574.

[9] R. Dwivedi, D. Dave, H. Naik, S. Singhal, R. Omer, P. Patel, B. Qian, Z. Wen, T. Shah, G. Morgan, R. Ranjan, Explainable ai (xai): Core ideas, techniques, and solutions, ACM Comput. Surv. 55 (2023). URL: https://doi.org/10.1145/3561048. doi:10.1145/3561048.

[10] D. Minh, H. X. Wang, Y. F. Li, T. N. Nguyen, Explainable artificial intelligence: a comprehensive review, Artificial Intelligence Review 55 (2022) 3503–3568. URL: https://doi.org/10.1007/s10462-021-10088-y. doi:10.1007/s10462-021-10088-y.

[11] F. Hinder, V. Vaquet, J. Brinkrolf, B. Hammer, Model-based explanations of concept drift, Neurocomputing 555 (2023) 126640.

[12] F. Bayram, B. S. Ahmed, A. Kassler, From concept drift to model degradation: An overview on performance-aware drift detectors, Knowledge-Based Systems 245 (2022) 108632. doi:https://doi.org/10.1016/j.knosys.2022.108632.

[13] S. Farquhar, Y. Gal, What'out-of-distribution'is and is not, in: Neurips ml safety workshop, 2022.

[14] R. N. Gemaque, A. F. J. Costa, R. Giusti, E. M. dos Santos, An overview of unsupervised drift detection methods, WIREs Data Mining and Knowledge Discovery 10 (2020) e1381. doi:https://doi.org/10.1002/widm.1381.

[15] P. Shen, Y. Ming, H. Li, J. Gao, W. Zhang, Unsupervised concept drift detectors: A survey, in: Advances in Natural Computation, Fuzzy Systems and Knowledge Discovery, 2023, pp. 1117–1124.

[16] J. Gama, P. Medas, G. Castillo, P. Rodrigues, Learning with drift detection, in: Advances in Artificial Intelligence, 2004, pp. 286–295.

[17] M. Baena-Garcıa, J. Del Campo-Ávila, R. Fidalgo, A. Bifet, R. Gavalda, R. Morales-Bueno, Early drift detection method, Fourth international workshop on knowledge discovery from data streams (2006).

[18] J. Gama, G. Castillo, Learning with local drift detection, in: Advanced Data Mining and Applications, Springer, 2006, pp. 42–55.

[19] I. Frías-Blanco, J. d. Campo-Ávila, G. Ramos-Jiménez, R. Morales-Bueno, A. Ortiz-Díaz, Y. Caballero-Mota, Online and non-parametric drift detection methods based on hoeffding's bounds, IEEE

Transactions on Knowledge and Data Engineering (2015). doi:10.1109/TKDE.2014.2345382.
[20] S. Rabanser, S. Günnemann, Z. C. Lipton, Failing loudly: An empirical study of methods for detecting dataset shift, in: Neural Information Processing Systems, 2018. URL: https://api.semanticscholar.org/CorpusID:53096511.
[21] S. Greco, T. Cerquitelli, Drift lens: Real-time unsupervised concept drift detection by evaluating per-label embedding distributions, in: 2021 International Conference on Data Mining Workshops (ICDMW), 2021, pp. 341–349. doi:10.1109/ICDMW53433.2021.00049.
[22] L. Bu, C. Alippi, D. Zhao, A pdf-free change detection test based on density difference estimation, IEEE Transactions on Neural Networks and Learning Systems 29 (2018) 324–334. doi:10.1109/TNNLS.2016.2619909.
[23] S. Greco, B. Vacchetti, D. Apiletti, T. Cerquitelli, Driftlens: A concept drift detection tool, in: Proceedings 27th International Conference on Extending Database Technology, EDBT 2024, Paestum, Italy, March 25 - March 28, OpenProceedings.org, 2024, pp. 806–809. URL: https://doi.org/10.48786/edbt.2024.75. doi:10.48786/edbt.2024.75.
[24] E. Lughofer, E. Weigl, W. Heidl, C. Eitzinger, T. Radauer, Drift detection in data stream classification without fully labelled instances, in: 2015 IEEE International Conference on Evolving and Adaptive Intelligent Systems (EAIS), 2015, pp. 1–8. doi:10.1109/EAIS.2015.7368802.
[25] A. Suprem, J. Arulraj, C. Pu, J. Ferreira, Odin: automated drift detection and recovery in video analytics, Proc. VLDB Endow. (2020). URL: https://doi.org/10.14778/3407790.3407837. doi:10.14778/3407790.3407837.
[26] M. Hushchyn, A. Ustyuzhanin, Generalization of change-point detection in time series data based on direct density ratio estimation, CoRR abs/2001.06386 (2020). URL: https://arxiv.org/abs/2001.06386. arXiv:2001.06386.
[27] K. Yamanishi, J.-i. Takeuchi, A unifying framework for detecting outliers and change points from non-stationary time series data, in: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '02, 2002. URL: https://doi.org/10.1145/775047.775148. doi:10.1145/775047.775148.
[28] O. Gözüaçık, A. Büyükçakır, H. Bonab, F. Can, Unsupervised concept drift detection with a discriminative classifier, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 2365–2368. URL: https://doi.org/10.1145/3357384.3358144. doi:10.1145/3357384.3358144.
[29] A. Liu, Y. Song, G. Zhang, J. Lu, Regional concept drift detection and density synchronized drift adaptation, 2017, pp. 2280–2286. doi:10.24963/ijcai.2017/317.
[30] S. Hido, T. Idé, H. Kashima, H. Kubo, H. Matsuzawa, Unsupervised change analysis using supervised learning, in: Advances in Knowledge Discovery and Data Mining, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 148–159.
[31] F. Hinder, V. Vaquet, J. Brinkrolf, A. Artelt, B. Hammer, Localization of concept drift: Identifying the drifting datapoints, in: 2022 International Joint Conference on Neural Networks (IJCNN), 2022, pp. 1–9. doi:10.1109/IJCNN55064.2022.9892374.
[32] G. I. Webb, R. Hyde, H. Cao, H. L. Nguyen, F. Petitjean, Characterizing concept drift, Data Mining and Knowledge Discovery 30 (2016) 964–994. URL: https://doi.org/10.1007/s10618-015-0448-4. doi:10.1007/s10618-015-0448-4.
[33] X. Wang, W. Chen, J. Xia, Z. Chen, D. Xu, X. Wu, M. Xu, T. Schreck, Conceptexplorer: Visual analysis of concept drifts in multi-source time-series data, in: 2020 IEEE Conference on Visual Analytics Science and Technology (VAST), 2020, pp. 1–11. doi:10.1109/VAST50239.2020.00006.
[34] K. B. Pratt, G. Tschapek, Visualizing concept drift, in: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '03, Association for Computing Machinery, New York, NY, USA, 2003, p. 735–740. URL: https://doi.org/10.1145/956750.956849. doi:10.1145/956750.956849.
[35] J. N. Adams, S. J. van Zelst, L. Quack, K. Hausmann, W. M. P. van der Aalst, T. Rose, A framework for explainable concept drift detection in process mining, in: Business Process Management,

Springer International Publishing, Cham, 2021, pp. 400–416.
[36] F. Hinder, V. Vaquet, B. Hammer, Feature-based analyses of concept drift, Neurocomputing 600 (2024) 127968. URL: https://www.sciencedirect.com/science/article/pii/S0925231224007392. doi:https://doi.org/10.1016/j.neucom.2024.127968.
[37] C. Duckworth, F. P. Chmiel, D. K. Burns, Z. D. Zlatev, N. M. White, T. W. V. Daniels, M. Kiuber, M. J. Boniface, Using explainable machine learning to characterise data drift and detect emergent health risks for emergency department admissions during covid-19, Scientific Reports 11 (2021) 23017. URL: https://doi.org/10.1038/s41598-021-02481-y. doi:10.1038/s41598-021-02481-y.
[38] Susnjak, Teo, Maddigan, Paula, Forecasting patient flows with pandemic induced concept drift using explainable machine learning, EPJ Data Sci. 12 (2023) 11. URL: https://doi.org/10.1140/epjds/s13688-023-00387-5. doi:10.1140/epjds/s13688-023-00387-5.
[39] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, Curran Associates Inc., Red Hook, NY, USA, 2017, p. 4768–4777.
[40] S. Greco, B. Vacchetti, D. Apiletti, T. Cerquitelli, Unsupervised concept drift detection from deep learning representations in real-time, IEEE Transactions on Knowledge and Data Engineering 37 (2025) 6232–6245. doi:10.1109/TKDE.2025.3593123.
[41] J. a. Gama, I. Žliobaitundefined, A. Bifet, M. Pechenizkiy, A. Bouchachia, A survey on concept drift adaptation, ACM Comput. Surv. 46 (2014). doi:10.1145/2523813.
[42] L. Yuan, H. Li, B. Xia, C. Gao, M. Liu, W. Yuan, X. You, Recent advances in concept drift adaptation methods for deep learning, in: L. D. Raedt (Ed.), Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22, International Joint Conferences on Artificial Intelligence Organization, 2022, pp. 5654–5661. URL: https://doi.org/10.24963/ijcai.2022/788. doi:10.24963/ijcai.2022/788, survey Track.
[43] Q. Xiang, L. Zi, X. Cong, Y. Wang, Concept drift adaptation methods under the deep learning framework: A literature review, Applied Sciences 13 (2023). URL: https://www.mdpi.com/2076-3417/13/11/6515. doi:10.3390/app13116515.
[44] M. Sundararajan, A. Taly, Q. Yan, Axiomatic attribution for deep networks, 2017. URL: https://arxiv.org/abs/1703.01365. arXiv:1703.01365.
[45] K. Simonyan, A. Vedaldi, A. Zisserman, Deep inside convolutional networks: Visualising image classification models and saliency maps, 2014. URL: https://arxiv.org/abs/1312.6034. arXiv:1312.6034.
[46] F. Ventura, S. Greco, D. Apiletti, T. Cerquitelli, Explaining deep convolutional models by measuring the influence of interpretable features in image classification, Data Mining and Knowledge Discovery 38 (2024) 3169–3226. URL: https://doi.org/10.1007/s10618-023-00915-x. doi:10.1007/s10618-023-00915-x.
[47] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, D. Batra, Grad-cam: Why did you say that?, 2017. URL: https://arxiv.org/abs/1611.07450. arXiv:1611.07450.
[48] J. Townsend, T. Kasioumis, H. Inakoshi, Eric: Extracting relations inferred from convolutions, in: Proceedings of the Asian Conference on Computer Vision, 2020.
[49] J. R. Quinlan, C4. 5: programs for machine learning, Elsevier, 2014.
[50] P. Padalkar, H. Wang, G. Gupta, Nesyfold: a framework for interpretable image classification, in: Proceedings of the AAAI Conference On Artificial Intelligence, volume 38, 2024, pp. 4378–4387.
[51] P. Padalkar, G. Gupta, Symbolic rule extraction from attention-guided sparse representations in vision transformers, arXiv preprint arXiv:2505.06745 (2025).
[52] R. F. Tate, Correlation between a discrete and a continuous variable. point-biserial correlation, The Annals of mathematical statistics 25 (1954) 603–607.
[53] Z. Liu, P. Luo, X. Wang, X. Tang, Deep learning face attributes in the wild, in: Proceedings of International Conference on Computer Vision (ICCV), 2015.