

# Interpretable Model Drift Detection

Pranoy Panda

Indian Institute of Technology  
Hyderabad, India

Vineeth N Balasubramanian  
Indian Institute of Technology  
Hyderabad, India

## ABSTRACT

Data in the real world often has an evolving distribution. Thus, machine learning models trained on such data get outdated over time. This phenomenon is called model drift. Knowledge of this drift serves two purposes: (i) Retain an accurate model and (ii) Discovery of knowledge or insights about change in the relationship between input features and output variable w.r.t. the model. Most existing works focus only on detecting model drift but offer no interpretability. In this work, we take a principled approach to study the problem of interpretable model drift detection from a risk perspective using a feature-interaction aware hypothesis testing framework, which enjoys guarantees on test power. The proposed framework is generic, i.e., it can be adapted to both classification and regression tasks. Experiments on several standard drift detection datasets show that our method is superior to existing interpretable methods (especially on real-world datasets) and on par with state-of-the-art black-box drift detection methods. We also quantitatively and qualitatively study the interpretability aspect including a case study on USENET2 dataset. We find our method focuses on model and drift sensitive features compared to baseline interpretable drift detectors.

### ACM Reference Format:

Pranoy Panda, Kancheti Sai Srinivas, Vineeth N Balasubramanian, and Gaurav Sinha. 2018. Interpretable Model Drift Detection. In *Proceedings of Joint International Conference on Data Science and Management of Data (CODS-COMAD)*. ACM, New York, NY, USA, 9 pages. <https://doi.org/3632410.3632434>

## 1 INTRODUCTION

A standard assumption in traditional supervised learning settings is that input data is sampled from a stationary distribution. In reality, however, many application domains – including finance, healthcare, energy informatics, and communications – generate data that evolve with time, i.e., they are non-stationary [12, 33, 34]. Such an evolution of data over time can make models learned through standard supervised learning techniques underperform, i.e., the decision boundary learned by the model drifts away from

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

CODS-COMAD, Jan 04–07, 2024, Bengaluru, KA, IND

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-XXXX-X/18/06...\$15.00

<https://doi.org/3632410.3632434>

Kancheti Sai Srinivas

Indian Institute of Technology  
Hyderabad, India

Gaurav Sinha

Microsoft Research  
Bengaluru, India

the actual decision boundary over time [26]. This phenomenon of model degradation is termed as *model drift*. Related settings have been studied under different names in literature – most primarily, as concept drift – which we discuss along with related work in Sec 2. To better understand model drift, consider a machine learning (ML) or neural network (NN) model trained on data points produced by a data-generating process. This generating process can change over time, due to evolving dynamics of a given application setting, resulting in a shift in the covariate distribution (a.k.a *covariate shift*), shift in feature-conditioned output/posterior distribution (a.k.a *posterior shift/real concept drift*), or both [20]. However, not all data shifts result in significant model performance degradation – i.e. some data shifts are benign w.r.t. the model (for e.g., data that lie far away from the decision boundary in the correct direction). Retraining the model in such scenarios could be unnecessary and rather increase the cost of model deployment.

Besides retraining when the model performance degrades, it is useful to understand such a drift in terms of input features. It can be helpful to deduce change in variables of importance, i.e. features that were predictive before the drift but not after, and vice versa. With the growing emphasis on interpretable models, such kinds of insights are helpful for users to understand which variable – say, user demand, geographic location or season – changed in such a drift. Thus, *interpretable model drift detection* is an important problem to be studied as it has a potentially wide range of applications including predictive maintenance [50], social media analysis [9, 36] and malware detection [24], where model drifts are common.

Existing methods in literature to detect model drift include KS test based adaptive WINdowing (KSWIN) [41], McDiarmid Drift Detection Method (MDDM) [39], Drift Detection Method (DDM) [19], Early Drift Detection Method (EDDM) [4] and Adaptive Windowing Algorithm (ADWIN) [5]. Although these methods detect model drift, they are not interpretable. The limited efforts that have considered interpretability in drift detection [15, 28] are restricted to drifts in the covariate space, which may not always lead to model drift. To the best of our knowledge, no holistic framework exists to address interpretable model drift detection. Existing post-hoc explainability methods are not intended for distribution shifts, and thus give unreliable explanations under drifts [29]. Therefore, in this work, we propose feature-inTeraction awaRe InterPretable mOdel Drift Detection (TRIPODD), a method that leverages hypothesis testing and model risk to detect model drift and simultaneously interpret it w.r.t. input features. Table 1 summarizes the comparison of TRIPODD with existing methods, and shows its usefulness and generalizability over earlier efforts.

**Table 1:** Comparison of our method with different model drift detectors in literature. Marginal [15] and Conditional [28] are covariate shift methods, which do not focus on model drift. ✓ and ✗ represent *True* and *False*, respectively.

Properties → Methods ↓	Focus on model drift	Feature-level interpretability	Relevant for classification task	Relevant for regression task	No assumptions on covariates
Marginal [15]	✗	✓	✓	✓	✓
Conditional [28]	✗	✓	✓	✓	✗
MDDM [39]	✓	✗	✓	✗	✓
DDM [19]	✓	✗	✓	✗	✓
ADWIN [5]	✓	✗	✓	✓	✓
KSWIN [41]	✓	✗	✓	✓	✓
TRIPODD (Ours)	✓	✓	✓	✓	✓

The proposed TRIPODD method adopts a first-principles approach and uses the base model’s empirical risk to directly study change in model performance on the prediction task w.r.t. a model class. Empirical risk minimization [46] has remained a key foundation of the field of machine learning, which is used to train ML and NN models; hence, defining change in decision boundary in terms of model risk is a natural choice. To attain feature-level interpretability of the drift, we formally define feature-sensitive model drift definition (Defn 3.2) and construct a hypothesis test around that definition, which is sensitive to feature interactions learned by the underlying NN model. This is then used for detecting and interpreting drifts on real-world datasets. Our key contributions are summarized below:

- We propose a new method, TRIPODD, for interpretable model drift detection. To the best of our knowledge, this is the first such work towards feature-interpretable model drift detection, paralleling the significant uptake of interpretability across application domains at this time.
- As TRIPODD uses only model risk to achieve this objective, it can be applied to both classification and regression tasks, thus making it a fairly generic method.
- We theoretically analyze the hypothesis testing framework underlying our method, and show that our proposed framework has guarantees on test power.
- We perform a comprehensive suite of experiments on 10 synthetic and 5 real-world datasets which show the superior interpretability of TRIPODD over well-known state-of-the-art baseline methods for the task, while performing at par or better in terms of model drift detection performance. We study interpretability both quantitatively and qualitatively to validate the usefulness of interpretations in our framework.

## 2 RELATED WORK

Distribution shift over time and its detection has been studied extensively in literature under different names [1, 20, 21, 31, 37, 42, 45]. The most popular among them is *concept drift* [20, 45]. Such a drift could occur due to change in covariate distribution (referred to as *virtual drift* or *covariate shift*), or change in the posterior distribution (referred to as *real concept drift* or *posterior shift*) [20], or both. *Dataset shift* [35] is another term used to capture such shift in distribution. In this work, we study model drift, which deals with deterioration of the model performance due to evolution in the data distribution. It does not directly fall into the above mentioned categories as data distribution shift (covariate shift or posterior

shift) need not always lead to significant model degradation. Below we give a summary of works in the space of concept drift detection methods and research efforts for interpreting drifts.

**Concept Drift Detection Methods:** Methods such as LSDD-CDT [7], Marginal [15] and Conditional Test [28] study covariate shift. Focusing on the covariate distribution can make these methods vulnerable to benign drifts in real-world data streams that do not change the model. These methods are hence known to generate many false positives when applied, thus limiting their usefulness in practice. On the other hand, posterior shift methods (also called real concept drift methods) such as MDDM [39], DDM [19], EDDM [4] and ADWIN [5] track misclassifications of a classifier and detect drift when the distribution of error changes, i.e. track change in posterior distribution by using cues from a given model. Such methods are relatively more robust to benign drifts in data streams. We study model drift instead, which is more contemporarily practical but does not fall into these categories as we take the model perspective instead of the data perspective.

**Interpretability in Drift Detection:** From an interpretability perspective, while most popular drift detection methods including MDDM [39], DDM [19], EDDM [4], SDDM [32], ADWIN [5] and KSWIN [41] are black-box methods, there have been efforts that attempt to detect drift and simultaneously understand different aspects of the drift that can provide a user with insights. These can be broadly categorized into visualization-based methods and feature-interpretable methods. *Visualization-based methods* provide insights by maps or plots which inform the user about the distribution change. For e.g., [47] studied drift using quantitative descriptions of drift in the marginal distributions, and then used marginal drift magnitudes between time periods to plot heat maps that gives insights of the drift. [40] developed a visualization tool that used parallel histograms to study concept drift that a user could visually inspect.

*Feature-interpretable methods* [14, 15, 28, 30] attempt to detect drift and simultaneously notify which features might be responsible for it. Marginal [15, 27] performs a feature-wise Kolmogorov-Smirnov (KS) test, while Conditional Test [28] performs a hypothesis test to check for a change in the distribution of each feature conditioned on the other input features. [28] addresses the adversarial drift detection problem in their work which is slightly different from the standard drift detection problem. [30] introduced a cellular automaton-based drift detector, where its cellular structure becomes a representation of the input feature space. However, this

work does not deal with model drift. While our method also falls in this category of feature-interpretable methods, our objective of model drift differs from these methods. Using model risk as a direct indicator allows us to apply our method in all kinds of prediction tasks, including classification and regression. A relatively older method [22] also uses a notion of model risk to study concept drift, but does not address interpretability. Our feature-sensitive model drift definition (Sec 4) allows us to model the drift in an interpretable manner & provide theoretical guarantees that are desirable for a hypothesis testing framework.

### 3 BACKGROUND AND PRELIMINARIES

*Notations.* We use capital letters to represent random variables and corresponding small letters to represent the values taken by these random variables. Bold-faced letters denote vectors or sets of variables. For each positive integer  $n$ , we denote the set  $\{1, \dots, n\}$  by  $[n]$  and the set  $\{m, m+1, \dots, n\}$  by  $[m, n]$ . By  $x \sim p(X)$ , we mean that  $x$  is obtained by sampling from the distribution  $p(X)$ . We represent our input covariates by  $X \in \mathcal{X}$  where  $\mathcal{X} \subset \mathbb{R}^d$ , and output variable as  $Y$ , which takes values in set  $\mathcal{Y}$ , such that  $\mathcal{Y} \subset \mathbb{R}$  for regression problems and  $\mathcal{Y} = [C]$  for classification problems ( $C$  is the number of classes). Let  $D_p = \{x_i, y_i\}_{i=1}^T \sim p(X, Y)$  and  $D_q = \{x_i, y_i\}_{i=T+1}^N \sim q(X, Y)$  be set of two samples and  $\mathcal{H}$  be some model class. The risk associated with a model  $h \in \mathcal{H}$  on distribution  $p(X, Y)$  is given by  $\mathcal{R}_p^L(h) = \mathbb{E}_{(x, y) \sim p(X, Y)} [L(h(x), y)]$ , for some loss function  $L$ . The corresponding empirical risk for sample  $D_p$  is given by  $\hat{\mathcal{R}}_{D_p}^L(h) = \frac{1}{T} \sum_{i=1}^T L(h(x_i), y_i)$ . Unless otherwise specified, we will assume the loss function to be some fixed unknown function and use  $\mathcal{R}_p(h)$ ,  $\hat{\mathcal{R}}_{D_p}(h)$  instead of  $\mathcal{R}_p^L(h)$ ,  $\hat{\mathcal{R}}_{D_p}^L(h)$  respectively. For each  $i \in [d]$ , we define  $e_i$  to be the vector with 1 in the  $i^{th}$  co-ordinate and 0 elsewhere. For any set  $S \subset [d]$  and vector  $x \in \mathbb{R}^d$ , by  $x \odot S$  we denote the orthogonal projection of  $x$  onto the subspace spanned by  $\{e_i, i \in S\}$  i.e.  $x \odot S$  matches  $x$  at all co-ordinates belonging to  $S$  and is 0 in all the other co-ordinates.

For any subset  $S \subset [d]$ , we define the subset-specific risk as  $\mathcal{R}_p^S(h) = \mathbb{E}_{(x, y) \sim p} [L(h(x \odot S), y)]$ . Finally, for each  $i \in [d]$ , we define a vector  $\Delta_p^i(h) \in \mathbb{R}^{2^d}$  that captures the change in risk when the  $i^{th}$  feature is added to all the other subsets of features. Using the index of all subsets  $S \subset [d]$  (in any fixed order), we formally define the co-ordinates of the  $2^d$  dimensional vector  $\Delta_p^i(h)$  as:

$$(\Delta_p^i(h))_S = \mathcal{R}_p^S(h) - \mathcal{R}_p^{S \cup \{i\}}(h)$$

Using these notations, we now define model drift and feature-sensitive model drift to study the problem of interpretable model drift detection in a principled manner. We note that our definitions can be used for both classification and regression settings since they depend only on the change in the risk of the model.

*Definition 3.1 (Model Drift).* Consider a stream of samples  $\{(x_t, y_t) : t = 1, 2, \dots\}$ . We say a model drift occurs at time  $t = T$ , if there exist two distributions  $p(X, Y), q(X, Y)$  on the joint variable  $(X, Y)$  and a model  $h$  trained on samples from  $p(X, Y)$  distribution, s.t.,

- (1)  $(x_t, y_t) \sim p(X, Y)$  for  $t \leq T$ ,
- (2)  $(x_t, y_t) \sim q(X, Y)$  for  $t > T$ , and
- (3)  $\mathcal{R}_p(h) \neq \mathcal{R}_q(h)$ .

The above definition says that there is a model drift if the true risk of the model  $h$  is different when the distribution changes from  $p(X, Y)$  to  $q(X, Y)$ . Change in risk indicates that the model may not perform well on samples from the new distribution  $q(X, Y)$ .

*Definition 3.2 (Feature-Sensitive Model Drift).* Consider a stream of samples  $\{(x_t, y_t) : t = 1, 2, \dots\}$ . We say that a feature-sensitive model drift occurs at time  $t = T$ , if there exist distributions  $p(X, Y), q(X, Y)$  on the joint variable  $(X, Y)$  and a model  $h$  trained on samples from  $p(X, Y)$ , such that:

- (1)  $(x_t, y_t) \sim p(X, Y)$  for  $t \leq T$ ,
- (2)  $(x_t, y_t) \sim q(X, Y)$  for  $t > T$ , and
- (3) There exists  $i \in [d]$ , such that  $\Delta_p^i(h) \neq \Delta_q^i(h)$ .

As mentioned earlier,  $\Delta_p^i(h)$  contains the change in subset specific risk  $\mathcal{R}_p^S(h)$  for all subsets  $S \subset [d]$ , when the  $i^{th}$  feature is added to it. Intuitively, it contains the “impact” of adding the  $i^{th}$  feature to other subsets of features, measured as a change in the subset specific risk. Thus, a feature sensitive drift occurs if, for some feature, this impact is different for  $t \leq T$  and  $t > T$ . Such an approach allows us to consider all possible feature interactions, and is hence reliable. One could view our approach as a first-principles approach premised on model risk and feature-level interpretability. We now describe our hypothesis testing framework for detecting feature-sensitive model drift.

### 4 TRIPODD: METHODOLOGY

We begin by defining the hypothesis testing framework which we build on Definition 3.2. We then define our test statistic which helps to conduct our hypothesis test on samples, followed by our overall methodology of TRIPODD including its algorithm. We also theoretically analyze the test power of our test statistic and show that it converges to 1 as the number of samples  $n \rightarrow \infty$ .

#### 4.1 Hypothesis Testing Framework

As stated earlier, our framework is directly built on top of the definition of feature-sensitive model drift definition (Defn 3.2), as it relates existence of model drift to input features. It states that the presence of such a drift requires at least one feature  $k$  and a subset of features  $S$  for which the change in subset-specific risk  $\mathcal{R}_p^S(h) - \mathcal{R}_p^{S \cup \{k\}}(h)$  and  $\mathcal{R}_q^S(h) - \mathcal{R}_q^{S \cup \{k\}}(h)$  are different. These features  $(k)$  have different effects on model risk for the two distributions  $p$  and  $q$  in Defn 3.2, and can thus be used as an interpretation of the drift. Our method leverages a hypothesis testing framework to identify these features.

*Definition 4.1 (Hypothesis Test).* The null hypothesis  $\mathbf{H}_0$  and alternate hypotheses  $\mathbf{H}_a$  for the effect of the  $k^{th}$  feature on model risk is given by:

$\mathbf{H}_0$ : For all subsets of features  $S \subseteq [d]$ ,

$$\mathcal{R}_p^S(h) - \mathcal{R}_p^{S \cup \{k\}}(h) = \mathcal{R}_q^S(h) - \mathcal{R}_q^{S \cup \{k\}}(h)$$

$\mathbf{H}_a$ :  $\exists S \subseteq [d]$ , such that,

$$\mathcal{R}_p^S(h) - \mathcal{R}_p^{S \cup \{k\}}(h) \neq \mathcal{R}_q^S(h) - \mathcal{R}_q^{S \cup \{k\}}(h)$$

We signal a drift if and only if  $\mathbf{H}_0$  is rejected for some feature  $k \in [d]$ . We note that the null hypothesis  $\mathbf{H}_0$  is true if and only if there is no feature-sensitive model drift, and,  $\mathbf{H}_a$  is true otherwise.

Let  $d^k(h) := \max_{S \subseteq F} |(\mathcal{R}_p^S(h) - \mathcal{R}_p^{S \cup \{k\}}(h)) - (\mathcal{R}_q^S(h) - \mathcal{R}_q^{S \cup \{k\}}(h))|$  and  $\hat{d}^k(h)$  be its sample estimate i.e.,  $\hat{d}^k(h) := \max_{S \subseteq F} |(\hat{\mathcal{R}}_{D_p}^S(h) - \hat{\mathcal{R}}_{D_p}^{S \cup \{k\}}(h)) - (\hat{\mathcal{R}}_{D_q}^S(h) - \hat{\mathcal{R}}_{D_q}^{S \cup \{k\}}(h))|$ . It is easy to see that  $d^k(h) > 0$  (strictly greater) if and only if the alternate hypothesis is true, otherwise it is 0. We define our test statistic as follows.

**Definition 4.2 (Test Statistic).** Given two streams of samples  $D_p = \{(\mathbf{x}_i, y_i) \sim p(\mathbf{X}, Y) : i \in [n]\}$  and  $D_q = \{(\mathbf{x}_j, y_j) \sim q(\mathbf{X}, Y) : j \in [n+1, 2n]\}$ , along with a model  $h$  trained on samples from  $p(\mathbf{X}, Y)$ , we define our sample test statistic  $\hat{c}_n^k(h)$  for the  $k^{th}$  feature as:

$$\hat{c}_n^k(h) := n\hat{d}^k(h) \quad (1)$$

The population counterpart is defined as  $c_n^k(h) := nd^k(h)$ .

One could view the quantity  $d^k(h)$  used in our test statistic above as similar to the Marginal Contribution Importance (MCI) score in [8] (or Shapley values [43] in terms of measuring feature contributions); however, MCI or Shapley values do not focus on model drift. Besides, our statistic and our corresponding hypothesis testing framework directly follow from our definitions in Defns 3.2 and 4.1 towards model drift detection, making this an equivalent first-principles approach for interpreting model drift detection.

## 4.2 Methodology

Given the test statistic defined in Defn 1, we now describe our methodology for drift detection. TRIPODD uses a sliding window procedure and maintains a reference window (samples from old distribution), new samples window (samples from current distribution) and a model trained on samples from old distribution. It compares the risk between reference ( $Z_R$ ) and new samples windows ( $Z_N$ ) for every feature by using the test statistic stated above (note that first  $n - \lfloor nr \rfloor$  samples of  $Z_N$  are used for the test to ensure risk is computed on  $Z_R$  &  $Z_N$  on an equal number of samples). In any hypothesis testing framework, it becomes important to understand the statistical significance of the test statistic values at a given point. To this end, we use thresholds evaluated using a bootstrapping procedure, briefly described below. If there exists at least one feature for which the computed test statistic is greater than the corresponding threshold, then it declares model drift and all features that result in a drift become part of the drift interpretation. The entire procedure for implementing TRIPODD is detailed in Algorithm 1. The *Performance* variable stores the model performance (accuracy in classification, or  $R^2$  value in regression) across the data stream.

**Bootstrap procedure:** (for  $K$  bootstraps & significance level  $\alpha$ ) In this procedure we merge and shuffle samples from  $Z_N$  and  $Z_R$ , and then pick  $K$  two-samples from this mixture. This simulates the null hypothesis. Finally, we calculate the test statistic for these  $K$  two-samples and return the  $(1 - \frac{\alpha}{d})$ -th quantile of the test statistic values as our threshold (we use Bonferroni correction as we perform multiple comparisons).

**Efficient Treatment of Subsets.** For dealing with datasets with a large number of features, following earlier efforts such as MCI [8] that leverage sampling techniques, we use the random sampling technique in [10] to reduce the computational overhead. A random set of permutations of features is sampled, for which our test statistic is computed. We show in our experiments and analysis (see

**Algorithm 1** TRIPODD (feature-inTeraction awaRe InterPretabLe mOdEl Drift Detection)

---

```

Require:  $\mathcal{H}, n, \alpha, r, K, \delta, \mathcal{Z} = \{z_t = (\mathbf{x}_t, y_t) : t = 1, 2, \dots\}$ 
1: Create empty list Performance  $\leftarrow \phi$ . ▷ Model performance across stream
2: Create empty list Interpretation  $\leftarrow \phi$ . ▷ Drift related features
3:  $\tilde{n} \leftarrow n - \lfloor nr \rfloor$  ▷ Effective window size
4: Initialize  $i \leftarrow n$ , and model  $h \leftarrow \text{GETMODEL}(\mathcal{H}, \{z_t : t \in \{1, \dots, \lfloor nr \rfloor\}\})$ 
5:  $Z_R = \{z_t : t \in \{1 + \lfloor nr \rfloor, \dots, n\}\}$ 
6: while True do
7:   Flag  $\leftarrow \text{False}$  ▷ Drift flag
8:    $Z_N = \{z_t : t \in [i + 1, \dots, i + n]\}$ .
9:   Compute  $\hat{c}_n^k(h), \forall k \in [d]$ , using  $Z_R, Z_N, h$  in Eq 1.
10:  Get thresholds  $(T_\alpha^1, \dots, T_\alpha^d) \leftarrow \text{BOOTSTRAP}(h, \alpha, K, Z_R, Z_N)$ 
11:  if for any  $k \in [d], \hat{c}_n^k(h) > T_\alpha^k$  then
12:    Flag  $\leftarrow \text{True}$ , Add all such  $k$ s to list Interpretation.
13:  end if
14:  if Flag = True then ▷ Drift detected
15:     $h \leftarrow \text{GETMODEL}(\mathcal{H}, \{z_t : t \in [i + 1, \dots, i + \lfloor nr \rfloor]\})$ 
16:     $Z_R \leftarrow \{z_t : t \in [i + \lfloor nr \rfloor + 1, \dots, i + n]\}$ 
17:    Add model perf of  $h$  on  $Z_R$  to the Performance list
18:     $i \leftarrow i + n$  ▷ Shift windows by  $n$ 
19:    Print the list Interpretation and reset it to  $\phi$ .
20:  else
21:    Add model perf of  $h$  on  $Z_N$  to Performance list
22:     $i \leftarrow i + \delta$  ▷ Shift windows by  $\delta$  if no drift detected
23:  end if
24: end while

```

---

Section 6 that our method's time complexity is significantly lesser than the sampling rates of the datasets typically used for model drift detection, making this application setting a relevant one for such a feature-interaction based approach.

## 4.3 Guarantees of Test Statistic

To show the goodness of the proposed test statistic, we theoretically analyze its test power and consistency. We use a bootstrap sampling approach from [16] to simulate the null hypothesis which addresses the consistency of our test. We now show that the power of our test converges i.e. for any threshold (corresponding to some significance level  $\alpha$ ), the probability that our statistic is larger than threshold tends to 1, when the alternate hypothesis is true. We formally state our results below and provided the proof in Appendix.

**THEOREM 4.3 (CONVERGENCE OF TEST POWER).** *The test power of the proposed hypothesis test in Definition 4.1 converges to the ideal value 1 under the alternate hypothesis, i.e. suppose the alternate hypothesis  $H_a$  is true for some feature  $k \in [d]$ , then, for any  $t > 0$ ,  $\lim_{n \rightarrow \infty} \mathbb{P}[\hat{c}_n^k(h) > t] = 1$ .*

## 5 EXPERIMENTS AND RESULTS

We study two aspects of TRIPODD – its model-drift detection capability and its interpretability. We comprehensively evaluate our method on an extensive suite of synthetic, semi-synthetic, and real-world datasets, and observe that TRIPODD provides a strong sense of interpretability while performing at par or

**Table 3:** Datasets used in our experiments.  $D$  = number of features; *Drift-type* indicates if drift is abrupt or gradual and the type of drift in the dataset – covariate shift (*cov*), posterior shift (*pos*) or a mixture of both (*mix*).

Type	Dataset Name	D	Task	Drift-type
Syn	Sine [19]	4	Class.	abrupt,mix
	Agrawal [2]	9	Class.	abrupt,mix
	Mixed [38]	6	Class.	abrupt,cov
	Aug-Mixed	6	Class.	abrupt,mix
	SEA [6]	3	Class.	abrupt,mix
	SEA-Gradual [6]	3	Class.	gradual,mix
	Hyperplane [6]	10	Class.	gradual,mix
Real	Friedmann [18]	4	Reg.	abrupt,mix
	Airlines [6]	7	Class.	unknown
	USENET2 [23]	100	Class.	unknown
	Electricity [23]	5	Class.	unknown
	Weather [17]	9	Class.	unknown
PowerSupply [11]	PowerSupply [11]	2	Class.	unknown
	Air Quality [13]	8	Reg.	unknown

**Table 2: Drift Detection Results:** We report Average Model Performance across the data stream ( $M$ ) for all datasets, and Precision ( $P_{det}$ ) + Recall ( $R_{det}$ ) for synthetic datasets with ground truth drift locations. ( $M$  = accuracy for classification;  $M = R^2$  value for regression). We compare TRIPODD with 2 interpretable (Marginal & Conditional) and 4 black-box drift detectors (KSWIN, MDDM, DDM, ADWIN) on 10 synthetic datasets and 5 real-world datasets. Interpretable methods are highlighted in gray. *AD (Always Drift)* for Marginal & Conditional indicates that these methods, since not designed for model drift, detected drift at every new window, making them ineffective for drift detection. Dataset variations with *imbalance* are added to show results with increased complexity of class imbalance. Higher is better for all metrics. Values in bold & underlined indicate best and second best.

Methods → Datasets ↓	TRIPODD (Ours)	Marginal	Conditional	KSWIN	MDDM	DDM	ADWIN
	M $P_{det}$ $R_{det}$	M $P_{det}$ $R_{det}$	M $P_{det}$ $R_{det}$	M $P_{det}$ $R_{det}$	M $P_{det}$ $R_{det}$	M $P_{det}$ $R_{det}$	M $P_{det}$ $R_{det}$
<i>Classification Task</i>							
Sine (balance)	<b>55.5</b> <b>1.0</b> <b>1.0</b>	42.2 0.0 0.0	42.2 0.0 0.0	51.3 <b>1.0</b> <b>1.0</b>	<b>55.2</b> <b>1.0</b> <b>1.0</b>	50.0 <b>1.0</b> <b>1.0</b>	<b>55.5</b> <b>1.0</b> <b>1.0</b>
Sine (imbalance)	55.0 <b>1.0</b> <b>1.0</b>	42.2 0.0 0.0	42.2 0.0 0.0	<b>59.4</b> <b>1.0</b> <b>1.0</b>	<b>59.0</b> <b>1.0</b> <b>1.0</b>	50.0 <b>1.0</b> 0.3	56.0 <b>1.0</b> <b>1.0</b>
Agrawal (balance)	<b>55.0</b> <b>1.0</b> <b>1.0</b>	<b>55.0</b> 0.8 <b>1.0</b>	<b>54.0</b> <b>1.0</b> <b>1.0</b>	50.3 <b>1.0</b> <b>1.0</b>	51.5 <b>1.0</b> <b>1.0</b>	50.0 <b>1.0</b> <b>1.0</b>	51.1 <b>1.0</b> <b>1.0</b>
Agrawal (imbalance)	<b>52.0</b> <b>1.0</b> <b>1.0</b>	52.0 0.8 0.8	52.0 <b>1.0</b> 0.5	48.0 0.8 <b>1.0</b>	52.0 <b>1.0</b> 0.8	45.0 <b>1.0</b> 0.3	<b>56.2</b> <b>1.0</b> <b>1.0</b>
Mixed	<b>99.2</b> 0.7 <b>1.0</b>	98.9 0.5 <b>1.0</b>	<b>99.2</b> 0.5 <b>1.0</b>	98.1 <b>1.0</b> 0.3	98.1 <b>1.0</b> 0.3	98.0 <b>1.0</b> 0.4	<u>99.0</u> 0.8 <b>1.0</b>
Aug-Mixed	<b>94.0</b> <b>1.0</b> <b>0.8</b>	90.0 0.5 0.6	90.0 0.4 0.6	<u>92.7</u> <b>1.0</b> 0.6	<u>92.7</u> <b>1.0</b> 0.6	90.0 <b>1.0</b> 0.4	92.0 <b>1.0</b> <b>0.8</b>
SEA	<b>92.0</b> <u>0.7</u> <b>1.0</b>	88.6 <b>1.0</b> 0.3	88.2 <b>1.0</b> 0.3	<u>90.0</u> <b>1.0</b> 0.7	<u>90.0</u> <b>1.0</b> 0.7	89.0 <b>1.0</b> 0.3	89.5 <b>1.0</b> 0.3
SEA-Gradual	<b>88.0</b> - -	87.3 - -	87.3 - -	<u>87.8</u> - -	87.3 - -	87.3 - -	87.3 - -
Hyperplane	<b>88.1</b> - -	86.6 - -	87.4 - -	<u>87.5</u> - -	86.7 - -	85.9 - -	<u>87.5</u> - -
Airlines	57.2 - -	AD - -	AD - -	<b>59.7</b> - -	56.7 - -	56.5 - -	<u>57.8</u> - -
Electricity	<b>77.6</b> - -	AD - -	AD - -	<u>74.7</u> - -	74.6 - -	73.4 - -	74.5 - -
Weather	<b>77.4</b> - -	AD - -	AD - -	75.6 - -	<b>77.4</b> - -	<b>77.4</b> - -	<b>72.1</b> - -
Powersupply	<b>72.6</b> - -	AD - -	AD - -	<u>72.3</u> - -	<u>72.3</u> - -	71.7 - -	72.0 - -
<i>Regression Task</i>							
Friedmann	<b>0.65</b> <u>0.40</u> <b>1.0</b>	0.60 <u>0.4</u> <b>1.0</b>	<u>0.64</u> 0.3 <b>1.0</b>	0.62 0.3 <b>1.0</b>	- - -	- - -	0.62 <b>0.8</b> <b>1.0</b>
Air Quality	<b>0.48</b> - -	AD - -	AD - -	<u>0.22</u> - -	- - -	- - -	<u>0.22</u> - -

better than existing state-of-the-art in drift detection capabilities.

### 5.1 Model Drift Detection

We begin with our study of drift detection performance of TRIPODD against existing methods.

**Datasets:** We evaluate TRIPODD on 10 synthetic datasets and 5 real-world datasets that are popularly used for drift detection [31], as listed in Table 3. TRIPODD is task-agnostic and can detect different model drifts including those caused by covariate shift, posterior shift and a mixture of both. We create two harder datasets – *Sine (imbalance)* and *Agrawal (imbalance)* by adding class imbalance to *Sine* & *Agrawal* respectively. We expect that data with class imbalance is challenging to drift detectors, as they may ignore drift pertaining to minority classes. We also added an *Aug-Mixed* dataset that has both covariate and posterior shift by extending the *Mixed* dataset that only has covariate shift.

**Baselines:** We compare our method against 6 well-known drift detection methods: KSWIN [41], MDDM [39], DDM [19], ADWIN [5], Marginal [15, 27] and Conditional [28]. MDDM and DDM can only detect drifts in the classification setting while KSWIN and ADWIN work in both classification and regression settings. All four methods do not offer feature-level interpretability for the drift. Marginal and Conditional are interpretable by design, but only look at covariate shift (as in Table 1).

**Metrics:** For synthetic datasets where the true drift time is known, following [48], we report precision ( $P_{det}$ ) & recall ( $R_{det}$ ) of detected drifts. True positive (TP), false positive (FP), and false negative (FN) are identified as follows: a detected drift is a TP if it is detected

within a small fixed time range (tolerance window) of the true drift location; FN refers to missing a drift within the fixed time range; FP is a detection outside the fixed time range or an extra detection in the fixed time range of the true drift location. The tolerance window for all synthetic datasets was chosen to be half of the detection window length. For real-world datasets where there is no ground truth drift localization, we follow [44] and other prior works in reporting average model performance computed across the data stream. We use average accuracy for classification problems and average  $R^2$  value for regression problems. In this metric, the practice followed is to retrain the model when a drift is detected, to maintain model performance. We report this metric also for synthetic datasets.

**Results:** Table 2 reports our results for drift detection. TRIPODD consistently shows the best model performance across all datasets. TRIPODD is consistently strong on all drift detection metrics when compared to baseline methods, especially considering it also provides interpretability (discussed later in this section). In particular, TRIPODD outperforms the interpretable baselines (*Marginal* and *Conditional*) on all synthetic datasets, and especially on datasets that contain posterior shifts (*Sine* & *Aug-Mixed*). For real-world datasets, *Marginal* and *Conditional* detect drifts at every new window since they were not designed to detect only model drift, making them ineffective in practice. [28] made a similar observation.

**Implementation Details.** The test statistic  $\hat{c}_n^k(h)$ (for a feature  $k$ ) is computed using  $\mathcal{Z}_R \sim p(X, Y)$ ,  $\mathcal{Z}_N \sim q(X, Y)$  and the model  $h$ . The model  $h \in \mathcal{H}$  is learned by minimizing empirical risk on samples drawn from the distribution  $p(X, Y)$ . To evaluate model

risk  $h$ , we use a held-out set of samples. Therefore,  $\mathcal{Z}_R$  samples are kept disjoint from the training samples for  $h$  to ensure valid estimation of risk. Given a set of  $n$  samples, we train a model on the first  $\lfloor nr \rfloor$  samples, and the reference window  $\mathcal{Z}_R$  contains the remaining  $n - \lfloor nr \rfloor$  samples.  $r$  defines the proportion of samples used for training the model to computing the test statistic.  $r = 0.8$  for all our experiments in this section across all methods, for fairness of comparison. A larger fraction of training samples results in a well trained model. We study the effect of varying  $r$  in Sec 6, Table 10. For the bootstrap procedure used in Algorithm 1, following [28], we use  $K = 100$  bootstraps and  $\alpha = 0.05$ . For more details on the bootstrap procedure please refer to [28]. We use default parameters suggested in the respective papers for all baselines.

As described in Sec 4.2, TRIPODD uses a moving window procedure to detect drift on the input data stream. We follow the protocol used by [28] in this context. When there is no drift, the new samples window is shifted by  $\delta = 50$ . For avg model performance, window size  $n = 1000$  for all datasets and  $n = 1500$  for measuring precision and recall in synthetic datasets. A larger window size is used for syn datasets considering the availability of data in this setting. This also allows learning better models across the methods when estimating  $P_{det}$  &  $R_{det}$ . Choosing the appropriate window size is a task dependent issue. If the data stream is prone to many drifts (e.g. if known from domain knowledge), the window size should be small, as a large window size would lead to a delay in drift detection. If the task at hand is known to have drifts rarely, a slightly larger window size could be used as that would improve the quality of the hypothesis testing framework.

Base model is a 2-layer neural network for all our experiments in this section. We use a 2-layer neural network since it is the simplest model which performs well on all datasets and allows for quick empirical evaluation. However, TRIPODD detect drift for all model classes. We perform ablation studies in Sec 6 to study the impact of different window sizes and model classes on our method's performance.  $\delta$  is generally much smaller than window size  $n$  to detect drifts close to their true locations. When a drift is detected, a batch of  $n$  samples is requested and a new model is trained on the first  $\lfloor nr \rfloor$  samples of that batch and reference window is updated with the remaining  $n - \lfloor nr \rfloor$  samples. New samples window is shifted by  $n$ .

## 5.2 Model Drift Interpretability

TRIPODD not just detects drifts, but also provides interpretability in terms of input features that are most attributed to the drift. In order to study this, we conduct three kinds of studies: (i) We construct new synthetic datasets with known feature attributions and study the precision and recall for the specific features; (ii) We adapt the commonly used occlusion-based metric for interpretability to this setting, and study the change in model performance on datasets studied in Sec 5.1; and (iii) We perform a qualitative case study on the real-world USENET2 dataset to study interpretability in particular. We compare our method against *Marginal* and *Conditional*, which are the interpretable baselines.

**Study on Synthetic Datasets:** We construct two synthetic datasets D1 & D2 with known feature attributions, and report precision and recall of identifying these features in Table 4. D1 is a binary classification dataset on 3 binary variables  $(x_1, x_2, x_3)$  with a single

drift. The decision rule pre-drift is  $y_{pre} = (x_1 \oplus x_2) \cup x_3$  which post-drift becomes  $y_{post} = x_1 \cup x_3$ . Here,  $y_{pre}$  depends on the pair  $(x_1, x_2)$ , whereas in  $y_{post}$  the pair is not important. Thus, there is a drift due to the change in the relationship between  $(x_1, x_2)$  and  $y$ . Similarly, D2 has 4 binary variables with a single drift caused to change in decision rule from  $(x_1 \wedge x_2)$  to  $(x_1 \wedge x_3)$ . The cause of drift is change in the relation of  $(x_2, x_3)$  and  $y$ . These datasets have non-linear prediction rules and are challenging for our interpretable baselines. Results in Table 4 show that TRIPODD is able to identify relevant features better than *Marginal* and *Conditional* when tested under non-benign shifts.

**Table 4: Precision ( $P$ ) and Recall ( $R$ ) for feature localization on synthetic datasets with known feature attribution**

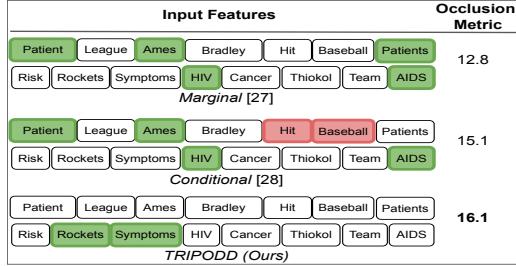
Metrics → Datasets ↓	$P$			$R$		
	TRIPODD	Marg	Cond	TRIPODD	Marg	Cond
D1	<b>1.0</b>	0	0	<b>1.0</b>	0	0
D2	<b>1.0</b>	<b>1.0</b>	<b>1.0</b>	<b>1.0</b>	<b>1.0</b>	0.5

**Study with Occlusion-based Metric:** Following the well-known strategy of occlusion in quantifying interpretability [3, 49], we occlude the features identified by a method, impute them with their average value and look at the change in accuracy drop. Let  $\Delta\mathcal{A}([d]) = A_{\mathcal{Z}_R}^{[d]}(h) - A_{\mathcal{Z}_N}^{[d]}(h)$  denote the difference in average accuracy of model  $h$  on reference samples  $\mathcal{Z}_R$  and new samples  $\mathcal{Z}_N$  when all  $d$  features are considered. When a drift is detected, we compute the importance of feature subset  $S$  as  $\sigma(S) = \Delta\mathcal{A}([d]) - \Delta\mathcal{A}([d] \setminus S)$ . One would expect a higher value of  $\sigma$  when the most important features are occluded. We define the occlusion metric across a data stream containing  $k$  drifts as  $\bar{\sigma} = \frac{1}{k} \sum_{i=1}^k \sigma(S_i)$ , where  $S_i$  is the feature subset selected as causing the drift. We show results for this study in Table 5. TRIPODD consistently outperforms other interpretable baseline methods.

**Table 5: Model drift leads to a drop in model accuracy, and features important to drift contribute the most to this drop in accuracy. We report our occlusion metric  $\bar{\sigma}$  in percentage. Higher is better.**

Datasets ↓ Methods →	TRIPODD	Marginal	Conditional
D1	<b>2.4</b>	0.0	0.0
D2	<b>60.0</b>	<b>60.0</b>	<b>60.0</b>
Sine (balance)	<b>32.0</b>	31.1	17.0
Sine (imbalance)	<b>25.3</b>	20.6	17.1
Mixed	<b>0.2</b>	-14.1	-11.2
Aug-Mixed	<b>7.6</b>	-6.0	2.0
SEA	<b>1.0</b>	-2.4	0.5
SEA-Gradual	<b>1.7</b>	0.2	-0.4
Hyperplane	<b>0.63</b>	0.05	-0.6

**Qualitative Case Study:** For this study, we use a real-world dataset USENET2 [25], which was obtained by asking people with different interests to label an email message as interesting or not interesting. There are three primary interest groups present in the dataset: *Space*, *Medicine* and *Baseball*. We use a subset of the dataset containing samples from two interest groups - *Space & Medicine*. Thus, this subset of the dataset contains a single drift between samples where people were interested in the *Space* category (size of sample set = 400) vs the samples where people were interested in the *Medicine* category (size of sample set = 400). As we know the cause of the drift in these two sets of samples (change in interest causing a



**Figure 1:** Qualitative Case Study for Drift Interpretability on USENET Dataset: Highlighted cells (green or red) are words related to drift w.r.t. each method. Green indicates a correct semantic connection to the drift, red indicates semantically irrelevant words. Our method selects relevant words with a higher sensitivity (as shown on the occlusion metric value). More details in Sec 5.2). Note: Bradley = baseball team & Ames = NASA center

change in posterior distribution), we can study the goodness of interpretations of the drift. We use this dataset since the feature semantics are easy to follow and do not require any background domain knowledge. Figure 1 shows the result. TRIPODD attributes model drift to words (here features) that are semantically related to the user interest categories that are involved in the drift i.e., *medicine* and *space*. Conditional attributes drift to a non-relevant interest category *baseball*. While Marginal also provides correct attributions, the occlusion metric values show higher attribution to the features identified by TRIPODD.

## 6 DISCUSSION AND ANALYSIS

**Effect of window size.** We study the effect of window size on drift detection and interpretability performance. The black-box drift detectors use adaptive windows or have recommended window sizes, thus we perform sensitivity analysis for drift detection of our method only.  $\delta = 50$  for window sizes greater than or equal to 1000, and  $\delta = 10$  for window sizes less than 1000 (for better sensitivity of drifts). We use a 2-layer neural network as our base model in this study. The results in Table 6 show that drift detection performance is not affected significantly by a change in window size. However, in a data stream, increasing the window size beyond a certain point can cause an increase in drift localization error (such as in the case of the Electricity dataset for a window size of 1500). We further conduct a similar study for drift interpretability in Table 8. The results show our method consistently performs better than the baseline methods.

**Table 6:** Sensitivity analysis on window size: Average model accuracy of TRIPODD for different window sizes.

Window Size →	750	1000	1250	1500
Datasets ↓				
Hyperplane	87.0	88.1	88.1	88.0
Electricity	77.7	77.6	77.1	76.5
Weather	77.1	77.4	77.5	77.1
Powersupply	72.4	72.6	73.5	72.4

**Effect of Model Class.** Here, we study the drift detection and interpretability performance of TRIPODD under different model classes and compare it with relevant baselines. We experiment with 3 different model classes: 2-layer (1024 & 512 neurons), 4-layer (1024, 512, 256 & 128 neurons) & 6-layer neural networks (1024, 512, 256, 128, 128 & 64 neurons) in Table 7. As can be seen from the table, TRIPODD performs at par or better than the black-box drift detectors, across

**Table 8:** Sensitivity analysis on win size: Occlusion metric comparison across different win sizes on the SEA dataset

Window Size →	750	1000	1250	1500
Methods ↓				
TRIPODD	<b>0.10</b>	<b>1.00</b>	<b>0.20</b>	<b>0.10</b>
Marginal	-2.10	-2.40	-0.27	<b>0.10</b>
Conditional	-3.70	0.50	0.00	0.00

model classes, while being interpretable at the same time. We further conduct a similar study for drift interpretability in Table 9. The results show our method consistently performs better than the baseline methods.

**Effect of r:** The ratio that defines the proportion of samples used for training the model to compute the test statistic in the reference window is referred to as  $r$ . Results reported in Table 2 were with the  $r$  value of 0.8. Here, in Table 10, we vary the value of  $r$  and study its impact on the drift detection performance of TRIPODD measured using the metric of average model performance across the data stream. It can be observed from Table 10 that the drift detection performance of TRIPODD stays approximately the same(within 1-2%) when  $r$  is varied indicating robustness of TRIPODD to  $r$  value. However, it is useful to note that if  $r$  is too high(very close to 1) then there would be very few samples left to calculate the test statistic and the risk estimate would be poor, leading to poor drift detection and interpretation. Similarly, if  $r$  is set to a very small value(very close to 0) then the base model would have to be learned on a small set of samples.

**Time Complexity.** Considering TRIPODD takes into account interactions of each feature with all possible subsets of features (similar to Shapley values or MCI scores), it incurs a time overhead to provide useful interpretations. However, the real-world benchmarks datasets used for drift detection have sampling rates in the order of hours (Electricity[23] dataset) or even days (Weather[17] dataset). Our method TRIPODD takes < 150 secs for Electricity dataset, and < 280 secs for Weather dataset, to perform the test on a window of size 1000 (both reference and new samples) with a 2-layer neural network. Considering the relative sampling rates of the datasets, our method is pseudo real-time, and is practically relevant & useful.

## 7 CONCLUSIONS

We propose TRIPODD to solve a contemporarily relevant, albeit less-studied problem of model drift detection. We define model drift in terms of input features and come up with a feature-wise hypothesis testing framework for detecting drift in an interpretable manner. We do not observe a trade-off between drift detection performance and interpretability – TRIPODD is interpretable and performs on par with state-of-the-art methods. Our method uses only model risk and can be applied to both classification and regression tasks, making it a general-purpose method. We conduct an extensive suite of experiments for drift detection and show that TRIPODD provides superior interpretability than existing methods, while performing at par or better than even black-box state-of-the-art methods for drift detection. We note that *model drift* can exist even when there is no *feature-sensitive model drift* - for eg. drift due to synergy

**Table 9:** Sensitivity analysis on model class: Occlusion metric comparison across model classes on SEA dataset. Here ‘k-layer’ refers to a k layer neural net.

Methods ↓	2-layer	4-layer	6-layer
TRIPODD	<b>1.0</b>	<b>0.1</b>	<b>0.3</b>
Marginal	-2.4	-1.62	-1.69
Conditional	0.5	0	0

Values of $r$ →	0.6	0.7	0.8
Datasets ↓			
Electricity	76.2	76.8	77.6
Weather	76.1	76.9	77.4

of a subset of features. Similar to existing feature-interpretable methods in literature [15, 28], we are interested in drifts that can be interpreted in terms of individual features in this work. Extending our method for multivariate model drift detection is a promising future direction.

## 8 APPENDIX

### Proof of theorem 4.3

**PROOF.** First we show that  $\hat{d}^k \xrightarrow{P} d^k$  asymptotically. Subsequently, we use this fact to find a bound on  $|\hat{c}_n^k - c_n^k|$ , which we use to derive a lower bound on the test power.

Let  $d^k(h) = \max_{S \subseteq F} |(\mathcal{R}_p^S(h) - \mathcal{R}_p^{S \cup \{k\}}(h)) - (\mathcal{R}_q^S(h) - \mathcal{R}_q^{S \cup \{k\}}(h))| = \max_{S \subseteq F} \Delta(S, k) \implies d^k(h) = \Delta(S^*, k)$ , where  $S^*$  is the subset which maximizes  $\Delta(S, k)$ . Similarly, we can define its sample counterpart as follows:  $\hat{d}^k(h) = \Delta_D(S^{**}, k)$ , where  $S^{**}$  is the subset which maximizes  $\Delta_D(S, k)$ . From now on we drop  $h$  from  $\hat{d}^k(h)$  &  $d^k(h)$  for simplicity of notation.

#### Convergence in Probability of $\hat{d}^k$ to $d^k$

$$\begin{aligned} \hat{d}^k - d^k &= \Delta_D(S^*, k) - \Delta(S^{**}, k) \leq \Delta_D(S^*, k) - \Delta(S^*, k) \\ &\leq \max_{S \subseteq F} |\Delta_D(S, k) - \Delta(S, k)| \end{aligned}$$

Similarly we can show that  $d^k - \hat{d}^k \leq \max_{S \subseteq F} |\Delta_D(S, k) - \Delta(S, k)|$ . Therefore, we can conclude:

$$|d^k - \hat{d}^k| \leq \max_{S \subseteq F} |\Delta_D(S, k) - \Delta(S, k)| \quad (2)$$

Now we find an upper bound of  $|\Delta_D(S, k) - \Delta(S, k)|$  with respect to error terms to ultimately upper bound the absolute difference between  $d^k$  and  $\hat{d}^k$ .

$$\begin{aligned} |\Delta_D(S, k) - \Delta(S, k)| &= \left| \underbrace{\left( \hat{\mathcal{R}}_{D_p}^S(h) - \hat{\mathcal{R}}_{D_p}^{S \cup \{k\}}(h) \right)}_{e_1} - \right. \\ &\quad \left. \underbrace{\left( \hat{\mathcal{R}}_{D_q}^S(h) - \hat{\mathcal{R}}_{D_q}^{S \cup \{k\}}(h) \right)}_{e_2} \right| - \left| \underbrace{\left( \mathcal{R}_p^S(h) - \mathcal{R}_p^{S \cup \{k\}}(h) \right)}_{e_3} - \right. \\ &\quad \left. \underbrace{\left( \mathcal{R}_q^S(h) - \mathcal{R}_q^{S \cup \{k\}}(h) \right)}_{e_4} \right| \\ &= |e_1 - e_2| - |e_3 - e_4| \leq |e_1 - e_3| + |e_4 - e_2| \end{aligned}$$

Substituting back the values,

$$\begin{aligned} |\hat{d}^k - d^k| &\leq \max_{S \subseteq F} |\hat{\mathcal{R}}_{D_p}^S(h) - \mathcal{R}_p^S(h)| + |\mathcal{R}_p^{S \cup \{k\}}(h) - \hat{\mathcal{R}}_{D_p}^{S \cup \{k\}}(h)| \\ &\quad + |\hat{\mathcal{R}}_{D_q}^S(h) - \mathcal{R}_q^S(h)| + |\mathcal{R}_q^{S \cup \{k\}}(h) - \hat{\mathcal{R}}_{D_q}^{S \cup \{k\}}(h)| \\ &= \max_{S \subseteq F} |\hat{\mathcal{R}}_{D_p}^S(h) - \mathcal{R}_p^S(h)| + \max_{S \subseteq F} |\mathcal{R}_p^{S \cup \{k\}}(h) - \hat{\mathcal{R}}_{D_p}^{S \cup \{k\}}(h)| \\ &\quad + \max_{S \subseteq F} |\hat{\mathcal{R}}_{D_q}^S(h) - \mathcal{R}_q^S(h)| + \max_{S \subseteq F} |\mathcal{R}_q^{S \cup \{k\}}(h) - \hat{\mathcal{R}}_{D_q}^{S \cup \{k\}}(h)| \end{aligned}$$

Now, we derive the lower bound on the probability of  $\hat{d}^k$  being different from  $d^k$ .

$$\begin{aligned} P(|\hat{d}^k - d^k| \leq \varepsilon) &\leq P(\max_{S \subseteq F} |\hat{\mathcal{R}}_{D_p}^S(h) - \mathcal{R}_p^S(h)| + \\ &\quad \max_{S \subseteq F} |\mathcal{R}_p^{S \cup \{k\}}(h) - \hat{\mathcal{R}}_{D_p}^{S \cup \{k\}}(h)| + \max_{S \subseteq F} |\hat{\mathcal{R}}_{D_q}^S(h) - \mathcal{R}_q^S(h)| \\ &\quad + \max_{S \subseteq F} |\mathcal{R}_q^{S \cup \{k\}}(h) - \hat{\mathcal{R}}_{D_q}^{S \cup \{k\}}(h)| \leq \varepsilon) \\ &\text{Using union bound of the following form -} \\ &P\left(\sum_{i=1}^m x_i \geq t\right) \leq \sum_{i=1}^m P\left(x_i > \frac{t}{m}\right) \text{ we get,} \\ P(|\hat{d}^k - d^k| \leq \varepsilon) &\geq 1 - P\left(\max_{S \subseteq F} |\hat{\mathcal{R}}_{D_p}^S(h) - \mathcal{R}_p^S(h)| \geq \frac{\varepsilon}{4}\right) \\ &\quad - P\left(\max_{S \subseteq F} |\mathcal{R}_p^{S \cup \{k\}}(h) - \hat{\mathcal{R}}_{D_p}^{S \cup \{k\}}(h)| \geq \frac{\varepsilon}{4}\right) \\ &\quad - P\left(\max_{S \subseteq F} |\hat{\mathcal{R}}_{D_q}^S(h) - \mathcal{R}_q^S(h)| \geq \frac{\varepsilon}{4}\right) \\ &\quad - P\left(\max_{S \subseteq F} |\mathcal{R}_q^{S \cup \{k\}}(h) - \hat{\mathcal{R}}_{D_q}^{S \cup \{k\}}(h)| \geq \frac{\varepsilon}{4}\right) \quad (3) \end{aligned}$$

We use union bound and Hoeffding's inequality to simplify each term in the RHS in the above equation.

$$\begin{aligned} P\left(\max_{S \subseteq F} |\hat{\mathcal{R}}_{D_p}^S(h) - \mathcal{R}_p^S(h)| \geq \frac{\varepsilon}{4}\right) &\leq P\left(\bigcup_{S \subseteq F} \{|\hat{\mathcal{R}}_{D_p}^S(h) - \mathcal{R}_p^S(h)| \geq \frac{\varepsilon}{4}\}\right) \\ &\leq \sum_{S \subseteq F} P\left(|\hat{\mathcal{R}}_{D_p}^S(h) - \mathcal{R}_p^S(h)| \geq \frac{\varepsilon}{4}\right) \leq 2^{|F|+1} \exp\left(\frac{-n\varepsilon^2}{8(M-m)^2}\right) \end{aligned}$$

Note:  $m \leq |\mathcal{R}_p^S(h) - \hat{\mathcal{R}}_{D_p}^S(h)| \leq M \forall S \subset F^1$

We get similar bounds for rest of the terms in RHS of Eq 3,

$$\begin{aligned} P\left(\max_{S \subseteq F} |\mathcal{R}_p^{S \cup \{k\}}(h) - \hat{\mathcal{R}}_{D_p}^{S \cup \{k\}}(h)| \geq \frac{\varepsilon}{4}\right) &\leq 2^{|F|+1} \exp\left(\frac{-n\varepsilon^2}{8(M-m)^2}\right) \\ P\left(\max_{S \subseteq F} |\hat{\mathcal{R}}_{D_q}^S(h) - \mathcal{R}_q^S(h)| \geq \frac{\varepsilon}{4}\right) &\leq 2^{|F|+1} \exp\left(\frac{-n\varepsilon^2}{8(M-m)^2}\right) \\ P\left(\max_{S \subseteq F} |\mathcal{R}_q^{S \cup \{k\}}(h) - \hat{\mathcal{R}}_{D_q}^{S \cup \{k\}}(h)| \geq \frac{\varepsilon}{4}\right) &\leq 2^{|F|+1} \exp\left(\frac{-n\varepsilon^2}{8(M-m)^2}\right) \end{aligned}$$

Thus, by using the above bounds we can simplify Equation 3 and also show convergence in probability of  $\hat{d}^k$  as follows:

$$P(|\hat{d}^k - d^k| \leq \varepsilon) \geq 1 - 2^{|F|+3} \exp\left(\frac{-n\varepsilon^2}{8(M-m)^2}\right) \quad (4)$$

$$\implies \lim_{n \rightarrow \infty} P(|\hat{d}^k - d^k| > \varepsilon) = 0 \quad \forall \varepsilon > 0$$

#### Convergence of Test Power

From Equation 4 we can infer the following:

$$P(|\hat{c}_n^k - c_n^k| \leq \varepsilon) \geq 1 - 2^{|F|+3} \exp\left(\frac{-\varepsilon^2}{8n(M-m)^2}\right)$$

The above bound can also be written in the following form:

$$\begin{aligned} \mathbb{P}(c_n^k - \varepsilon \leq \hat{c}_n^k \leq c_n^k + \varepsilon) &\geq 1 - 2^{|F|+3} \exp\left(\frac{-\varepsilon^2}{8n(M-m)^2}\right) \\ \implies \mathbb{P}(\hat{c}_n^k \geq c_n^k - \varepsilon) &\geq 1 - 2^{|F|+3} \exp\left(\frac{-\varepsilon^2}{8n(M-m)^2}\right) \end{aligned}$$

We now perform a simple reparameterization by substituting  $t = c_n^k - \varepsilon$  to get a lower bound on the test power.

$$\mathbb{P}(\hat{c}_n^k \geq t) \geq 1 - 2^{|F|+3} \exp\left(\frac{-(c_n^k - t)^2}{8n(M-m)^2}\right) = 1 - 2^{|F|+3} \exp\left(\frac{-n(d^k - \frac{t}{n})^2}{8(M-m)^2}\right)$$

Under the alternate hypothesis, it is easy to show that  $d^k > 0$ , therefore we conclude that,  $\lim_{n \rightarrow \infty} P(\hat{c}_n^k \geq t) = 1 \quad \forall t > 0$   $\square$

**A discussion on Interpretability of TRIPODD:** A drift can occur due to a combination of many features, but where no individual feature flags our hypothesis test. In this event, our method will not detect a drift, and will incur a false-negative. We cannot study the interpretability in this case, since no features are flagged as causing the drift. However this is a rather special case, and in general for tabular datasets with semantic, uncorrelated, features we expect drift to be ‘caused’ by one or many distinct features. In case of correlated features, our method can be easily extended by grouping the correlated features in a single feature. Our occlusion metric deals with the complementary case: In the event that a drift detection algorithm flags a drift, how relevant are the detected features? This metric does not consider the cases when the algorithm does not flag a drift. Our algorithm has a high recall for many datasets, indicating that there is no sacrifice in detection performance to ensure interpretability.

<sup>1</sup>This can be achieved with bounded loss functions. We assume this to be a practically reasonable assumption as in practice the loss values do not go out of bounds for models whose training does not diverge.

## REFERENCES

- [1] Supriya Agrahari and Anil Kumar Singh. 2021. Concept Drift Detection in Data Stream Mining: A literature review. *Journal of King Saud University-Computer and Information Sciences* (2021).
- [2] Rakesh Agrawal, Tomasz Imielinski, and Arun Swami. 1993. Database mining: A performance perspective. *IEEE transactions on knowledge and data engineering* 5, 6 (1993), 914–925.
- [3] Marco Ancona, Cengiz Oztireli, and Markus Gross. 2019. Explaining Deep Neural Networks with a Polynomial Time Algorithm for Shapley Value Approximation. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 272–281. <https://proceedings.mlr.press/v97/ancona19a.html>
- [4] Manuel Baena-García, José del Campo-Ávila, Raúl Fidalgo, Albert Bifet, R Gavalda, and Rafael Morales-Bueno. 2006. Early drift detection method. In *Fourth international workshop on knowledge discovery from data streams*, Vol. 6. 77–86.
- [5] Albert Bifet and Ricard Gavalda. 2007. Learning from time-changing data with adaptive windowing. In *Proceedings of the 2007 SIAM international conference on data mining*. SIAM, 443–448.
- [6] Albert Bifet, Geoff Holmes, Bernhard Pfahringer, Philipp Kranen, Hardy Kremer, Timm Jansen, and Thomas Seidl. 2010. Moa: Massive online analysis, a framework for stream classification and clustering. In *Proceedings of the first workshop on applications of pattern analysis*. PMLR, 44–50.
- [7] Li Bu, Cesare Alippi, and Dongbin Zhao. 2016. A pdf-free change detection test based on density difference estimation. *IEEE transactions on neural networks and learning systems* 29, 2 (2016), 324–334.
- [8] Amnon Catav, Boyang Fu, Yazeed Zoabi, Ahuva Libi Weiss Meilik, Noam Shomron, Jason Ernst, Sriram Sankararaman, and Ran Gilad-Bachrach. 2021. Marginal Contribution Feature Importance—an Axiomatic Approach for Explaining Data. In *International Conference on Machine Learning*. PMLR, 1324–1335.
- [9] Joana Costa, Catarina Silva, Mário Antunes, and Bernardete Ribeiro. 2014. Concept drift awareness in twitter streams. In *International Conference on Machine Learning and Applications*. IEEE, 294–299.
- [10] Ian Covert, Scott M Lundberg, and Su-In Lee. 2020. Understanding global feature contributions with additive importance measures. *Advances in Neural Information Processing Systems* 33 (2020), 17212–17223.
- [11] Hoang Anh Dau, Anthony Bagnall, Kavell Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. 2019. The UCR time series archive. *IEEE/CAA Journal of Automatica Sinica* 6, 6 (2019), 1293–1305.
- [12] Sharon E Davis, Thomas A Lasko, Guanhua Chen, and Michael E Matheny. 2017. Calibration drift among regression and machine learning models for hospital mortality. In *AMIA Annual Symposium Proceedings*, Vol. 2017. American Medical Informatics Association, 625.
- [13] Saverio De Vito, Ettore Massera, Marco Piga, Luca Martinotto, and Girolamo Di Francia. 2008. On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario. *Sensors and Actuators B: Chemical* 129, 2 (2008), 750–757.
- [14] Jaka Demšar and Zoran Bosnić. 2018. Detecting concept drift in data streams using model explanation. *Expert Systems with Applications* 92 (2018), 546–559.
- [15] Denis Moreira dos Reis, Peter Flach, Stan Matwin, and Gustavo Batista. 2016. Fast unsupervised online drift detection using incremental kolmogorov-smirnov test. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1545–1554.
- [16] Bradley Efron. 1992. Bootstrap methods: another look at the jackknife. In *Breakthroughs in statistics*. Springer, 569–593.
- [17] Ryan Elwell and Robi Polikar. 2011. Incremental learning of concept drift in nonstationary environments. *IEEE Transactions on Neural Networks* 22, 10 (2011), 1517–1531.
- [18] Jerome H Friedman. 1991. Multivariate adaptive regression splines. *The annals of statistics* 19, 1 (1991), 1–67.
- [19] Joao Gama, Pedro Medas, Gladys Castillo, and Pedro Rodrigues. 2004. Learning with drift detection. In *Brazilian symposium on artificial intelligence*. Springer, 286–295.
- [20] João Gama, Indré Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouacharia. 2014. A survey on concept drift adaptation. *ACM computing surveys (CSUR)* 46, 4 (2014), 1–37.
- [21] Paulo M Gonçalves Jr, Silas GT de Carvalho Santos, Roberto SM Barros, and Davi CL Vieira. 2014. A comparative study on concept drift detectors. *Expert Systems with Applications* 41, 18 (2014), 8144–8156.
- [22] Maayan Harel, Shie Mannor, Ran El-Yaniv, and Koby Crammer. 2014. Concept drift detection through resampling. In *International conference on machine learning*. PMLR, 1009–1017.
- [23] Michael Harries and New South Wales. 1999. Splice-2 comparative evaluation: Electricity pricing. (1999).
- [24] Roberto Jordaney, Kumar Sharad, Santanu K Dash, Zhi Wang, Davide Papini, Ilia Nouretdinov, and Lorenzo Cavallaro. 2017. Transcend: Detecting concept drift in malware classification models. In *USENIX Security Symposium (USENIX Security 17)*. 625–642.
- [25] Ioannis Katakis, Grigoris Tsoumakas, and Ioannis Vlahavas. 2008. An ensemble of classifiers for coping with recurring contexts in data streams. In *ECAI 2008*. IOS Press, 763–764.
- [26] Imen Khamassi, Moamar Sayed-Mouchaweh, Moez Hammami, and Khaled Ghédira. 2018. Discussion and review on evolving data streams and concept drift adapting. *Evolving systems* 9, 1 (2018), 1–23.
- [27] Daniel Kifer, Shai Ben-David, and Johannes Gehrke. 2004. Detecting change in data streams. In *VLDB*, Vol. 4. Toronto, Canada, 180–191.
- [28] Sean Kulinski, Saurabh Bagchi, and David I Inouye. 2020. Feature Shift Detection: Localizing Which Features Have Shifted via Conditional Distribution Tests.. In *Neural Information Processing Systems*.
- [29] Himabindu Lakkaraju, Nino Arsov, and Osbert Bastani. 2020. Robust and stable black box explanations. In *International Conference on Machine Learning*. PMLR, 5628–5638.
- [30] Jesus L Lobo, Javier Del Ser, Eneko Osaba, Albert Bifet, and Francisco Herrera. 2021. CURIE: a cellular automaton for concept drift detection. *Data Mining and Knowledge Discovery* 35, 6 (2021), 2655–2678.
- [31] Jie Lu, Anjin Liu, Fan Dong, Feng Gu, Joao Gama, and Guangquan Zhang. 2018. Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering* 31, 12 (2018), 2346–2363.
- [32] S. Micevska, A. Awad, and S. Sakr. 2021. SDDM: An Interpretable Statistical Concept Drift Detection Method for Data Streams. *Journal of Intelligent Information Systems* 56 (2021), 459–484. <https://doi.org/10.1007/s10844-020-00634-5>
- [33] Lilian Minne, Saeid Esfami, Nicolette De Keizer, Evert De Jonge, Sophia E De Rooij, and Ameen Abu-Hanna. 2012. Effect of changes over time in the performance of a customized SAPS-II model on the quality of care assessment. *Intensive care medicine* 38, 1 (2012), 40–46.
- [34] Karel GM Moons, Andre Pascal Kengne, Diederick E Grobbee, Patrick Royston, Yvonne Vergouwe, Douglas G Altman, and Mark Woodward. 2012. Risk prediction models: II. External validation, model updating, and impact assessment. *Heart* 98, 9 (2012), 691–698.
- [35] Jose G Moreno-Torres, Troy Raeder, Rocío Alaiz-Rodríguez, Nitesh V Chawla, and Francisco Herrera. 2012. A unifying view on dataset shift in classification. *Pattern recognition* 45, 1 (2012), 521–530.
- [36] Martin Müller and Marcel Salathé. 2020. Addressing machine learning concept drift reveals declining vaccine sentiment during the COVID-19 pandemic. *arXiv preprint arXiv:2012.02197* (2020).
- [37] Megha Ashok Patil, Sunil Kumar, Sandeep Kumar, and Muskan Garg. 2021. Concept Drift Detection for Social Media: A Survey. In *2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*. IEEE, 12–16.
- [38] Ali Pesaranghader, Herna L Viktor, and Eric Paquet. 2016. A framework for classification in data streams using multi-strategy learning. In *International conference on discovery science*. Springer, 341–355.
- [39] Ali Pesaranghader, Herna L Viktor, and Eric Paquet. 2018. McDiarmid drift detection methods for evolving data streams. In *International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–9.
- [40] Kevin B Pratt and Gleb Tschapek. 2003. Visualizing concept drift. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. 735–740.
- [41] Christoph Raab, Moritz Heusinger, and Frank-Michael Schleif. 2020. Reactive soft prototype computing for concept drift streams. *Neurocomputing* 416 (2020), 340–351.
- [42] Denise Maria Vecino Sato, Sheila Cristiana De Freitas, Jean Paul Barddal, and Edson Emilio Scalabrin. 2021. A survey on concept drift in process mining. *ACM Computing Surveys (CSUR)* 54, 9 (2021), 1–38.
- [43] LS Shapley. 1953. A value for n-person games. Contributions to the theory of games, , 307–318 pages.
- [44] Ashraf Tahmasbi, Ellango Jothimurugesan, Srikanta Tirthapura, and Phillip B Gibbons. 2020. DriftSurf: A Risk-competitive Learning Algorithm under Concept Drift. *arXiv preprint arXiv:2003.06508* (2020).
- [45] Alexey Tsymbal. 2004. The problem of concept drift: definitions and related work. *Computer Science Department, Trinity College Dublin* 106, 2 (2004), 58.
- [46] Vladimir Vapnik. 1991. Principles of risk minimization for learning theory. *Advances in neural information processing systems* 4 (1991).
- [47] Geoffrey I Webb, Loong Kuan Lee, Bart Goethals, and François Petitjean. 2018. Analyzing concept drift and shift from sample data. *Data Mining and Knowledge Discovery* 32, 5 (2018), 1179–1199.
- [48] Shujian Yu, Xiaoyang Wang, and José C Principe. 2018. Request-and-reverify: Hierarchical hypothesis testing for concept drift detection with expensive labels. *arXiv preprint arXiv:1806.10131* (2018).
- [49] Matthew D. Zeiler and Rob Fergus. 2013. Visualizing and Understanding Convolutional Networks. In *European Conference on Computer Vision*.
- [50] Jan Zenisek, Florian Holzinger, and Michael Affenzeller. 2019. Machine learning based concept drift detection for predictive maintenance. *Computers & Industrial Engineering* 137 (2019), 106031.