Interpretable Model Drift Detection

Pranoy Panda [1] **Sai Srinivas Kancheti** [1] **Vineeth N Balasubramanian** [1] **Gaurav Sinha** [2]

Abstract

Data in the real world often has an evolving data 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 Thus, machine learning models trained on such data get outdated over time.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 is applicable for 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. A qualitative study on interpretability, conducted on the USENET2 dataset, shows that our method focuses on relevant features. Also, our method achieves sparse interpretations compared to the baseline interpretable drift detection methods on the USENET2 dataset.

1. Introduction

It is a standard assumption in traditional supervised learning settings 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 evolves with time i.e., they are non-stationary (Moons et al., 2012; Davis et al., 2017; Minne et al., 2012). 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 the actual decision boundary over time (Khamassi et al., 2018). 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 6.

Fig 1 illustrates the difference between model drift and related settings. Consider a machine learning model (base model h in the figure) trained on data points sampled from an initial distribution or data generating process (Fig 1a).

[1]Indian Institute of Technology, Hyderabad [2]Adobe Research, Bangalore, India. Correspondence to: Pranoy Panda <cs20mtech12002@iith.ac.in>.

Workshop on Interpretable ML in Healthcare at International Conference on Machine Learning (ICML). Copyright 2022 by the author(s).

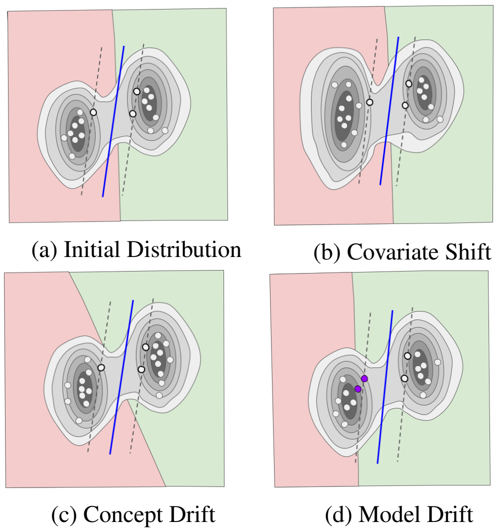


Figure 1: (a) shows the initial distribution where two axes represents two features, the red and green half-spaces denote the true posterior distribution and the blue line represents our base model h (such as a max-margin classifier), currently with no misclassifications. The circles denote data samples; circles colored white denote they are classified correctly by h. In (b), we see there is change in input feature distribution but no change in the true posterior distribution. h performance has also not changed i.e. number of misclassifications remains the same. In (c), we observe that the boundary between the two half-spaces has changed i.e. the true posterior distribution has shifted. But the base model h's performance has still not changed as no samples are misclassified. In (d), we observe that the half-space boundary has shifted towards the left, indicating a change in true posterior. But this time the base model no longer preserves its prior performance as it misclassifies 2 samples, shown as purple circles. Thus, retraining of the base model is required only in the final case i.e. when there is *model drift*.

This generating process can change over time resulting in shift in the input distribution, feature-conditioned output distribution (or posterior distribution), or both. Such shifts are typically studied as covariate shift when there is change in input distribution alone, or as concept drift for the other two cases (change in posterior is also sometimes called real concept drift) (Gama et al., 2014). However, not all such shifts cause the base model's performance to degrade, as shown in Fig 1b and c. Fig 1d shows a scenario when model starts misclassifying after the distribution shift, necessitating a detection of drift. Model drift detection can be even more useful in current times when machine learning models are extensively used across applications, and the user may not have access to the underlying true data ditribution but only to the model and a small set of data points. Detecting model drift helps inform a user about a distribution shift which potentially necessitates model retraining in order to retain model performance over time.

Existing methods in literature that addressed model drift include the McDiarmid Drift Detection Method (MDDM, state-of-the-art) (Pesaranghader et al., 2018), Drift Detection Method (DDM) (Gama et al., 2004), Early Drift Detection Method (EDDM) (Baena-Garcıa et al., 2006) and Adaptive Windowing Algorithm (ADWIN) (Bifet & Gavalda, 2007), which detected drift using model predictions. In this work, we use a model's empirical risk to directly study change in model performance on the prediction task w.r.t. a model class. Models belonging to different model classes could behave differently under the same distribution shift; for e.g., a less complex model class may not need to be retrained for certain subtle changes of distribution shift, while a more complex one may need to be. Moreover and importantly, none of the aforementioned methods offer ante-hoc interpretability i.e., they are black-box model drift detection methods. Understanding the reason behind model drift not only helps in retraining the model better, but also gives insights into the change in the underlying relationship between input and output variables w.r.t. the model. An intrinsically *interpretable model drift detection* method can be of immense value for applications such as predictive maintenance (Zenisek et al., 2019), understanding trends in social media (Costa et al., 2014; Müller & Salathé, 2020) and malware detection (Jordaney et al., 2017). While there have been some efforts in interpreting drift in general (Kulinski et al., 2020; Žliobaite, 2010), these methods are restricted to just drifts in feature space (covariate shift), which may not always lead to model drift.

In this work, we propose a feature-inTeraction awaRe InterPretable mOdel Drift Detection method (TRIPODD) to detect model drift in an interpretable manner, using a hypothesis testing framework. To this end, we first define feature-sensitive model drift using model risk. Machine learning models are learned by minimizing empirical risk (Vapnik, 1991), so defining the event of a change in decision boundary in terms of risk is natural. Most existing methods (MDDM, DDM, EDDM, ADWIN) use model misclassification rate with some heuristic to detect drift, we instead use model risk for this purpose. The proposed hypothesis testing framework enjoys guarantees on the consistency and power of the test statistic, which is desirable. The test statistic captures feature interactions, making it suitable for real-world datasets. Our empirical results on a mixture of real and synthetic datasets show the efficacy of our method against state-of-the-art baselines. We also show interpretability results to showcase the utility of our method. The key contributions of our work can be summarized as follows:

- We propose a new method, TRIPODD, for interpretable model drift detection using a feature interaction-aware hypothesis testing framework. To the best of our knowledge, this is the first work on generic feature-interpretable model drift, and can have many use cases considering that machine learning models are increasingly deployed in application domains that incur drift. As TRIPODD uses only model risk to achieve this objective, it can be applied to both classification and regression tasks, thus making it more generic than existing interpretable methods.
- We show that our proposed hypothesis testing framework has guarantees on test power.
- Our empirical studies across 3 synthetic and 4 real-world datasets show that TRIPODD performs better or at par with current state-of-the-art methods while being interpretable simultaneously. We also analyze the interpretations given by our method, and show that it provides sparse interpretations without correlated features.

2. 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 $\{1, \ldots, n\}$ by $[n]$. Expectation of random variable X is denoted by $\mathbb{E}[X]$. Let $p(X)$ be a probability distribution over random variable X. By $x \sim p(X)$, we mean that x is obtained by sampling from the distribution $p(X)$. We represent our input covariates by $\boldsymbol{X} \in \mathcal{X}$ 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 $D_q = \{x_i, y_i\}_{i=T}^{N} \sim q(X, Y)$ be set of two samples and $\mathcal{H}$ be some model class. The risk associated with a model $h \in \mathcal{H}$ on distribution $p(\boldsymbol{X}, Y)$ is given by $\mathcal{R}_p^L(h) = \mathbb{E}_{(\boldsymbol{x},y)\sim p(\boldsymbol{X},Y)}[L(h(x), y)]$, for some loss function L. The corresponding empirical risk for sample D_p

is given by $\hat{\mathcal{R}}^L_{D_p}(h) = \frac{1}{T}\sum_{i=1}^{T} L(h(x_i), y_i)$. 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 $\mathcal{R}^L_p(h), \hat{\mathcal{R}}^L_{D_p}(h)$ respectively. For each $i \in [d]$, we define $\boldsymbol{e}_i$ to be the vector with 1 in the i^{th} co-ordinate and 0 elsewhere. For any set $S \subset [d]$ and vector $\boldsymbol{x} \in \mathbb{R}^d$, by $\boldsymbol{x} \odot S$ we denote the orthogonal projection of $\boldsymbol{x}$ onto the subspace spanned by $\{\boldsymbol{e}_i, i \in S\}$ i.e. $\boldsymbol{x} \odot S$ matches $\boldsymbol{x}$ at all co-ordinates belonging to S and is 0 in all the co-ordinates. For any subset $S \subset [d]$, we defined the subset specific risk as $\mathcal{R}^S_p(h) = \mathbb{E}_{(\boldsymbol{x},y)\sim p}[L(h(x \odot S), y)]$. Finally, for each $i \in [d]$, we define a vector $\Delta^i_p(h) \in \mathbb{R}^{2^d}$ 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 $\Delta^i_p(h)$ as,

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

Next, using these notations, we 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[1] of the model.

Definition 2.1 (Model Drift)**.** Consider a stream of samples $\{(\boldsymbol{x}_t, y_t) : t = 1, 2, \ldots\}$. We say that a model drift occurs at time $t = T$, if there exist two distributions $p(\boldsymbol{X}, Y), q(\boldsymbol{X}, Y)$ on the joint variable $(\boldsymbol{X}, Y)$ and a model h trained on samples from $p(\boldsymbol{X}, Y)$ distribution, such that,

1. $(\boldsymbol{x}_t, y_t) \sim p(\boldsymbol{X}, Y)$ for $t \leq T$,
2. $(\boldsymbol{x}_t, y_t) \sim q(\boldsymbol{X}, Y)$ 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 there is distribution change from $p(\boldsymbol{X}, Y)$ to $q(\boldsymbol{X}, Y)$. Change in risk indicates that the model may not perform as well on samples from the new distribution $q(\boldsymbol{X}, Y)$.

Definition 2.2 (Feature Sensitive Model Drift)**.** Consider a stream of samples $\{(\boldsymbol{x}_t, y_t) : t = 1, 2, \ldots\}$. We say that a feature sensitive model drift occurs at time $t = T$, if there exist distributions $p(\boldsymbol{X}, Y), q(\boldsymbol{X}, Y)$ on the joint variable $(\boldsymbol{X}, Y)$ and a model h trained on samples from $p(\boldsymbol{X}, Y)$, such that,

1. $(\boldsymbol{x}_t, y_t) \sim p(\boldsymbol{X}, Y)$ for $t \leq T$,
2. $(\boldsymbol{x}_t, y_t) \sim q(\boldsymbol{X}, Y)$ for $t > T$, and
3. There exists $i \in [d]$, such that $\Delta^i_p(h) \neq \Delta^i_q(h)$.

[1]in case of Definition 2.2 it depends on risk restricted to subsets of features

As mentioned earlier, $\Delta^i_p(h)$ contains the change in subset specific risk $\mathcal{R}^S_p(h)$ 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 change in the subset specific risk. Thus, the above definition says that a feature sensitive drift occurs if for some feature this impact is different for $t \leq T$ and $t > T$. An important question to address here is why we consider the impact (of adding the i^{th} feature) for every possible subset of features. The reason is that considering all feature subsets enables us to take care of all possible interactions between the features, analogous to how Shapley values (Shapley, 1952) account for feature interactions in the attribution problem. We elaborate more on this in the Appendix. In the next section we describe a hypothesis testing framework for detecting feature sensitive model drift. In fact, our test statistic in Definition 3.2 has close resemblance to Marginal Contribution Importance (Catav et al., 2021) which was proposed as a solution to issues with Shapley values under the presence of correlated features for feature importance.

3. Feature-Interaction Aware Hypothesis Testing Framework for Interpretable Model Drift Detection

In this section we explain our proposed method TRIPODD. First we start with defining the hypothesis testing framework which follows directly from Definition 2.2. Then we define our test statistic which helps to conduct our hypothesis test on samples. Next, we present the methodology of TRIPODD wherein we state and briefly explain the algorithm. Finally, we end the section with an analysis on the test power of our test statistic showing that it converges to 1 as the number of samples $n \to \infty$.

3.1. Hypothesis Testing Framework

To detect model drift in an interpretable manner, we use the feature sensitive model drift definition as it relates existence of model drift to input features. That is, if there is such a drift then there exists at least one feature k and a subset of features S for which the change in subset specific risk $\mathcal{R}^S_p(h) - \mathcal{R}^{S\cup\{k\}}_p(h)$ and $\mathcal{R}^S_q(h) - \mathcal{R}^{S\cup\{k\}}_q(h)$ are different. As mentioned earlier such features (k) have different impact on the risk for the two distributions and can be used as an interpretation of the drift. Our algorithm performs hypothesis tests (defined next) and searches for these features.

Definition 3.1 (Hypothesis)**.** The hypothesis test corresponding to the k^{th} feature has the following null hypothesis $\mathbf{H}_0$ and alternate hypotheses $\mathbf{H}_a$.

$\mathbf{H}_0$: For all $S \subseteq F$,

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

$\mathbf{H}_a$: $\exists S \subseteq F$, 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 $k \in [d]$. Please 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.

Definition 3.2 (Test Statistic)**.** Given two streams of samples $\{(\boldsymbol{x}_i, y_i) \sim p(\mathbf{X}, Y) : i \in [n]\}$ and $\{(\boldsymbol{x}_j, y_j) \sim q(\mathbf{X}, Y) : j \in [n]\}$, along with a model h trained on samples from $p(Y, \boldsymbol{X})$ distribution, we define our sample test statistic for the k^{th} feature as:

$$\widehat{c}_n^k(h) = n\widehat{d}^k(h) \tag{1}$$

where $\widehat{d}^k(h) = \max_{S\subseteq F} |(\mathcal{R}_{D_p}^S(h) - \mathcal{R}_{D_p}^{S\cup\{k\}}(h)) - (\mathcal{R}_{D_q}^S(h) - \mathcal{R}_{D_q}^{S\cup\{k\}}(h))|$. The population counterpart is given by $c_n^k(h) = nd^k(h)$, where $d^k(h) = \max_{S\subseteq F} |(\mathcal{R}_p^S(h) - \mathcal{R}_p^{S\cup\{k\}}(h)) - (\mathcal{R}_q^S(h) - \mathcal{R}_q^{S\cup\{k\}}(h))|$

It is easy to show that $d^k(h) > 0$(strictly greater) *iff* the alternate hypothesis is true. Otherwise it is 0.

Relation to MCI(Catav et al., 2021): $d^k(h)$ in the proposed test statistic has a form similar to the Marginal Contribution Importance(MCI)(Catav et al., 2021) work.

3.2. Methodology

The entire procedure for implementing TRIPODD is detailed in Algorithm 1. TRIPODD uses a sliding window procedure and maintains a reference window(samples from old distribution), new samples window(samples from distribution at current time step) and a model trained on samples from old distribution. It compares risk between old and new samples windows for every feature by using the test statistic(definition 3.2). To understand if the computed test statistics are statistically significant it compares them to thresholds evaluated using a bootstrapping procedure. 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 such features become part of the drift interpretation.

Algorithm 1 TRIPODD (feature-inTeraction awaRe InterPretable mOdel Drift Detection)

```
Require: $\mathcal{H}, n, \alpha, r, K, \delta, \mathcal{Z} = \{\boldsymbol{z}_t = (\boldsymbol{x}_t, y_t) : t = 1, 2, \dots\}$
  $\tilde{n} \leftarrow n - \lfloor nr \rfloor$                ▷ Effective window size
Initialize $i \leftarrow n$, and model $h \leftarrow$ GetModel($\mathcal{H}, \{\boldsymbol{z}_t : t \in \{1, \dots, \lfloor nr \rfloor\}$)
$\mathcal{Z}_R = \{\boldsymbol{z}_t : t \in \{1 + \lfloor nr \rfloor, \dots, n\}\}$.
Create empty list Interpretation $\leftarrow \phi$.
while $True$ do
    Flag $\leftarrow False$                ▷ Drift Flag
    $\mathcal{Z}_N = \{\boldsymbol{z}_t : t \in [i, \dots, i + \tilde{n}]$.
    Compute $c_n^k(h), \forall k \in [d]$, using $\mathcal{Z}_R, \mathcal{Z}_N, h$ in Equation 1.
    Get thresholds $(T_\alpha^1, \dots, T_\alpha^d) \leftarrow$ Bootstrap($h, \alpha, K, \mathcal{Z}_R, \mathcal{Z}_N$)
    if for any $k \in [d]$, $c_n^k(h) > T_\alpha^k$ then
        Flag $\leftarrow True$, Add all such $k$s to the list Interpretation.
    end if
    ▷ Get new Model and reference window when drift detected
    if Flag $= True$ then
        $h \leftarrow$ GetModel($\mathcal{H}, \{\boldsymbol{z}_t : t \in [i, \dots, i + \lfloor nr \rfloor]$)
        $\mathcal{Z}_R \leftarrow \{\boldsymbol{z}_t : t \in [i + \lfloor nr \rfloor + 1, \dots, i + n]\}$
        $i \leftarrow i + n$          ▷ Shift windows by $n$
        Print the list Interpretation and reset it to $\phi$.
    else
        $i \leftarrow i + \delta$      ▷ Shift windows by $\delta$ if no drift detected
    end if
end while=0
```

Here we provided a slightly more detailed description of TRIPODD. TRIPODD uses a iterative window based procedure for detecting feature sensitive model drift on the input data stream $\mathcal{Z} = \{\boldsymbol{z}_t = (\boldsymbol{x}_t, y_t) : t = 1, 2, \dots\}$ by using the hypothesis testing framework described in Section 3.1. In the first iteration, a batch of n samples is fetched from the data stream $\mathcal{Z}$. A model $h \in \mathcal{H}$ is trained using the first $\lfloor nr \rfloor$ samples from this batch. r is the ratio that defines the proportion of samples used for training the model to computing the test statistic(input to algorithm 1). The rest $\tilde{n} = n - \lfloor nr \rfloor$ samples in the batch are stored in the *reference window* $\mathcal{Z}_R$. After this the algorithm is in detection mode i.e. it collects new samples and checks for model drift. For doing so a new set of $\tilde{n}$ samples are fetched from the data stream and are stored in the *new samples* window $\mathcal{Z}_N$. Please note, $\tilde{n}$ is the effective window size for our test, which is much smaller than n. Then the test statistic $\widehat{c}_n^k(h)$ is computed using $\mathcal{Z}_R, \mathcal{Z}_N$ and h for all $k \in [d]$ to check for model drift w.r.t. samples in $\mathcal{Z}_N$ and model h. To perform the test at a particular significance level α, we compute the thresholds $T_\alpha^k(\forall k \in [d])$ using a bootstrapping scheme followed by bonferroni correction(as we perform multiple comparisons). The bootstrapping procedure is explained in the Appendix. After the thresholds have been computed, the test statistic is compared with the thresholds, in a feature-wise manner. If there exists at least one feature for which the test statistic is greater than the threshold then TRIPODD declares a drift. Furthermore, all the features for which the test statistic exceeds the threshold become part of the interpretation for the drift. After a drift is detected, a new batch of n samples $\{\boldsymbol{z}_t : t \in [n, \ldots, 2n]\}$ is fetched from the data stream. Like before, the first $\lfloor nr \rfloor$ samples of this batch is used to get an updated model and the rest $\tilde{n}$ samples become the new $\mathcal{Z}_R$. Again, similar to before, next set of $\tilde{n}$ samples $\{\boldsymbol{z}_t : t \in [2n, \ldots, 2n + \tilde{n}]\}$ in the data stream become $\mathcal{Z}_N$. If there is no drift detected, then the new samples window $\mathcal{Z}_N$ is shifted by δ amount. δ is an input to the TRIPODD algorithm.

3.3. Guarantees of Test Statistic

To show the goodness of the proposed test statistic we analyze its test power and consistency. We use a bootstrap sampling approach from (Efron, 1992) to simulate the null hypothesis which takes care of consistency of our test. We now show that the power of our test converges i.e. for any threshold(corresponding to some significance level α), the probability that statistic is larger than threshold tends to 1, when the alternate hypothesis is true. We state the theorem for the above claim below and provide the proof in the Appendix.

Theorem 3.3 (Convergence of Test Power)**.** *The test power of our proposed test converges to the ideal value 1 under the alternate hypothesis. That is, suppose the alternate hypothesis H_a is true for some feature $k \in [d]$, then, for any $t > 0$, $\lim_{n\to\infty} \mathbb{P}[\hat{c}_n^k(h) > t] = 1$*

4. Experiments and Results

We conducted experiments across synthetic and real-world datasets to study the usefulness of the proposed method. We begin by describing our experimental setup – in particular, the datasets, evaluation metrics, baselines and implementation details.

Datasets. Following (Lu et al., 2018), we used the same datasets that are popularly used for drift detection, as summarized in Table 1. For more details on the datasets please refer to the Appendix.

Type	Name	Total samples	Dim	Prediction Task
Syn	Hyperplane (Bifet et al., 2010)	20000	10	Class.
	Mixed (Pesaranghader et al., 2016)	20000	6	Class.
	Friedmann (Friedman, 1991)	20000	4	Reg.
Real	USENET2* (Katakis et al., 2008)	1200	27	Class.
	Electricity (Harries & Wales, 1999)	45312	5	Class.
	Weather (Elwell & Polikar, 2011)	18159	9	Class.
	Air Quality (De Vito et al., 2008)	7532	8	Reg.

Table 1: Dataset Details (Dim = Dimensions; Class. = Classification task, Reg. = Regression Task)

Evaluation Metrics. Real-world streaming datasets generally do not have information on the true drift locations, therefore, following prior work (Tahmasbi et al., 2020), the underlying model's average performance (accuracy for classification problems and mean squared error for regression problems) is used as a proxy for measuring goodness of drift detection. The intuition behind this is that if a model is retrained at the true drift location, then the average model performance will suffer the least as we retrain/fine-tune the model before too much performance degradation happens. The model performance is always evaluated on the new samples window that it receives at every time step, and the average is computed over the entire data stream.

Implementation Details. *Hypothesis Test Details:* The test statistic $\hat{c}_n^k(h)$(for a feature k) is computed using $\mathcal{Z}_R \sim p(\boldsymbol{X}, Y)$, $\mathcal{Z}_N \sim q(\boldsymbol{X}, Y)$ and the model h. The model $h \in \mathcal{H}$ is learned by minimizing the empirical risk on samples drawn from the distribution $p(\boldsymbol{X}, Y)$. To evaluate risk of model h, we must 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. So, 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 is the ratio that defines the proportion of samples used for training the model to computing the test statistic. $r = 0.8$ for all our experiments across all the methods, for fairness of comparison. We perform an ablation study on the value of r and report it in the Appendix. For the bootstrap procedure used in Algorithm 1, we use $K = 50$ bootstraps. For more details about the bootstrap procedure, please see the Appendix.

Moving Window Scheme: As described in Sec 3.2, TRIPODD uses a moving window procedure to detect drift on the input data stream. When there is no drift, both the reference and new samples window is shifted by δ amount. For all our experiments in this section, we fix $\delta = 50$, window size $n = 1000$, and the base model as a 2-layer neural network. We perform ablation studies in Sec 5 to study the impact of different window sizes and model classes on our method's performance. δ is generally much smaller than the window size n to detect drifts close to their true locations. When there is a drift, a batch of n samples is requested and a new model is trained on the first subsequent $\lfloor nr \rfloor$ samples and the reference window is updated with the remaining $n - \lfloor nr \rfloor$ samples. Also, the new samples window is shifted by n.

Baselines. We evaluate TRIPODD in both classification and regression settings. (To the best of our knowledge, ours is the first work that operates both on classification and regression settings.) For the classification setting, we compare our method against 3 well-known supervised drift detection methods(black-box): MDDM (Pesaranghader et al., 2018)(state-of-the-art), DDM (Gama et al., 2004) and ADWIN (Bifet & Gavalda, 2007), and two interpretable drift detection methods: Marginal (Žliobaite, 2010) and Conditional Test (Kulinski et al., 2020) (denoted as Marg and Cond in the results). As there are fewer methods in the space

	Ours	MDDM	DDM	ADWIN	Marg	Cond
Mixed	**99.2**	98.5	98.1	98.0	98.9	**99.2**
Hyp	87.5	86.7	85.9	87.5	87.7	**88.0**
Elec	**74.9**	74.6	73.4	74.5	NA	NA
Weat	**77.8**	77.5	77.4	72.1	NA	NA

Table 2: Average classification accuracy comparison with baseline methods (NA denotes that respective method detects drifts at all time steps)

of regression, we compare our method with a recent method: KSWIN (Raab et al., 2020). For all baseline methods, we use the default parameters suggested in the respective papers. We use the `scikit-multiflow` open-source library (Montiel et al., 2018) for using ADWIN, DDM and KSWIN methods.

Detection Results. Classification Setting: Table 2 presents the results of our experiments on the classification setting. We use 2 synthetic (Mixed, Hyperplane) and 2 popular real-world datasets (Electricity, Weather) for these studies. Columns 3-5 show performance of supervised black-box

	Ours	KSWIN
Freidmann	**21.2**	31.0
Air Quality	**23.9**	38.3

Table 3: Avg MSE comparison with a recent black-box method

baseline methods; the last two columns refer to interpretable drift detection methods. As can be seen, our method performs better than the baseline methods w.r.t. the average classification accuracy metric of the base model. On the synthetic datasets, all the methods perform similarly. However, on the real-world datasets, the covariate shift detection methods (Marginal (Žliobaite, 2010) and Conditional Test (Kulinski et al., 2020)) detect drift at all time steps. A plausible reason for this is that natural data streams have benign drifts in the covariate space, and the covariate shift detection methods capture them leading to this problem. In other words, it is difficult for drift detection methods operating in covariate space to distinguish between benign and non-benign drifts in natural settings (a similar conclusion was also made in (Kulinski et al., 2020)). TRIPODD on the other hand focuses on model risk, and is hence robust to benign drifts in real-world data streams, since model performance is generally not affected due to benign changes. Hence, our method is robust to such changes.

Regression Setting: Table 3 shows comparison of TRIPODD with one supervised black-box baseline method KSWIN. We use 1 synthetic (Friedmann) and 1 real-world dataset (Air Quality) to perform the comparison. The results indicate that TRIPODD performs better than the baseline methods w.r.t. the average Mean Squared Error (MSE) metric of the base model while being interpretable at the same time.

Interpretability in Classification Setting - We consider a real-world dataset USENET2 to demonstrate our interpretability result in the classification setting. We use this dataset because the shifts in this dataset can be understood w.r.t. its features without any specific domain knowledge. The USENET2* dataset is constructed by asking users to label an email message as interesting or not interesting, based on their interest. People with different interests are asked to label the emails, which results in drifts in the dataset. There are three primary interest groups present in the dataset: *Space*, *Medicine* and *Baseball*. The dataset consists of 5 time segments; after each time segment, there is a shift in user interest which causes the drift. We show interpretability results for one drift in Fig 2. As shown in the figure, TRIPODD is able to attribute 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 Test(Kulinski et al., 2020) on the other hand attributes drift to a non-relevant interest category *baseball*. The Marginal method is correctly able to attribute feature importance to drift-relevant words, but finds many correlated words. TRIPODD finds sparse explanations which is desirable as discussed in explainability literature (Crabbé & Van Der Schaar, 2021; Chang et al., 2020). Additionally, we include results for a user study on interpretations on the USENET2 dataset in the Appendix.

Interpretability in Regression Setting - In order to demonstrate the interpretable nature of TRIPODD in the regression setting, we use the Friedmann dataset (Friedman, 1991). Ours is the only interpretable drift detection method for this prediction task. In the Friedmann dataset, the drift is caused due to change in distribution of individual input features. Drift is present at 5 locations in the data stream and at each drift location it is caused due to the first, third and fifth feature. Table 4 shows the features indicating model drift w.r.t. TRIPODD and the true drift causing features. As can be observed from Table 4, TRIPODD is able to detect all the drift-causing features. It has some false positives at later times, which we hypothesize could be due to accumulation of errors over time in model risk computations. Studying the effect of our method on long-range data streams may be an interesting future direction of this work.

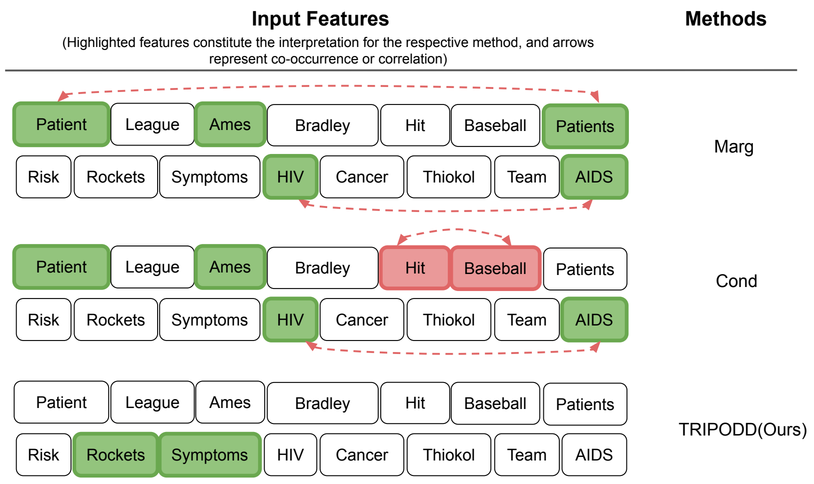

Input Features
(Highlighted features constitute the interpretation for the respective method, and arrows represent co-occurrence or correlation)
Methods
Patient
League
Ames
Bradley
Hit
Baseball
Patients
Risk
Rockets
Symptoms
HIV
Cancer
Thiokol
Team
AIDS
Marg
Cond
TRIPODD(Ours)

Figure 2: This example from USENET2 dataset shows drift due to shift in user interest from *Medicine* to *Space* categories. The highlighted cells are the words found to be most attributed(features that indicated drift) for detecting drift in the respective methods. Green-colored cells indicate words that also match the drift from common understanding i.e. either they belong to medicine or space category. Red-colored cells indicate words that are semantically irrelevant. Red arrows indicate correlations between such words. Evidently, our method selects relevant words, as well as avoids correlation in such concepts resulting in sparse interpretations. *Bradley* = baseball team. Note that baselines used here are covariate shift detection methods (not model drift, where we are the first), thus interpretations provided by them reflect change in feature space which may or may not lead to model drift (as in Fig 1).

Drift locs → Method ↓	2000	3000	5000	7500	16000
TRIPODD	**1,3,5**	**1,3,5**	**1,3,4,5**	**1,3,4,5**	**1,3,4,5**

Table 4: This table shows the features signalling model drift w.r.t. TRIPODD for each drift location in the Friedmann dataset; True drift causing features = (1,3,5); Blue color indicates true positive w.r.t. feature localization and red color indicates false positive.

Time Complexity. Considering TRIPODD takes into account interactions of each feature with all possible subsets of features (similar to Shapley values (Shapely, 1953) or MCI scores (Catav et al., 2021)), 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(Harries & Wales, 1999) dataset) or even days (Weather(Elwell & Polikar, 2011) dataset). Our method TRIPODD takes $\sim$ 100 secs for Electricity dataset, and $\sim$ 400 secs for Weather dataset, to perform the test on a window of size 1500 (both reference and new samples) with a 4-layer neural network. Considering the relative sampling rates of the datasets, our method is pseudo real-time, and is practically relevant and useful. For a more detailed discussion, please see Appendix.

5. Discussion and Analysis

In this section, we analyze and study different aspects of the proposed method, including choices of hyperparameters used in our earlier experiments. We use the classification task for these studies.

Correctness of Detected Drifts: In Section 4 we reported results on 4 popular real-world and 3 synthetic datasets. To further highlight the usefulness of our proposed method, we use two additional synthetic datasets where the location of drift is known apriori. Therefore, we can study the correctness of the detected drifts in a direct way for such datasets, i.e. we use precision and recall of detected drifts along with the proxy metric of average model performance. We define true positive (TP), false positive (FP) and false negative (FN) in our context as follows: TP is a detection that is made within a small fixed time range of the true drift location; FN refers to missing a detection within the fixed time range; FP is a detection outside the fixed time range (around the true drift location) or an extra detection in the time range of the true drift location. The max time range for each dataset is chosen to be the length of the detection window; all methods use the same time range. We choose to experiment on these two datasets because they covered both covariate and posterior shift based drifts, thus providing a lot of diversity. *Aug-Mixed* dataset is an augmentation of the Mixed dataset. It contains one covariate shift and two posterior shifts. As can be seen from Table 5, our method is able to detect all the drifts (recall = 1, albeit with some false positives), whereas Marginal and Conditional (as defined in Sec 4) cannot detect posterior shifts and hence have lower

Methods → Datasets ↓	Ours (P/R/A)	Marg (P/R/A)	Cond (P/R/A)
Aug-Mixed	**0.5/1.0/91.2**	**0.5**/0.6/85.5	0.4/0.6/87.6
Sine	**0.8/1.0/57.1**	0/0/42.2	0/0/42.2

Table 5: Precision(P), Recall(R) and Average classification accuracy(A) comparison with interpretable baseline methods. Window size = 1000, base model = 4 layer neural network

Window Size → Dataset ↓	750	1000	1250	1500
Hyperplane	87.0	87.5	88.1	88.0
Electricity	74.9	74.9	75.1	76.1
Weather	74.6	77.8	76.7	76.0

Table 6: Average classification accuracy of TRIPODD for different window sizes

recall. *Sine* dataset involves drifts in the target concept or the true labeling function, resulting in model degradation. Marg and Cond cannot detect them as covariate distribution is unchanged. Our method uses model risk, and is thus able to detect all types of drift that changes model performance.

Effect of window size. Here, we study the effect of window size on drift detection performance (on the same average classification accuracy metric used earlier). $\delta = 50$ 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 an increase in window size improves drift detection performance. This is expected as our proposed hypothesis test becomes more reliable with increase in sample size, since the model risk estimate becomes more accurate.

Effect of Model Class. Here, we study the drift detection performance of TRIPODD under different model classes. We experiment with 4 different model classes: logistic regression, 2-layer neural network, 4-layer neural network and a 6-layer neural network. Details about the model specifications are provided in the Appendix. Table 7 shows the starting base model accuracy and average base model accuracy over the respective data stream, for each base model class. It can be observed that the gap between the start and the final average accuracy remains within 1-2% for the Electricity dataset across the model types, and similarly it is within 2-4% for the Weather dataset. Given that these are real-world data streams which span over years in time scale, it can be inferred that TRIPODD is able to retain base model performance across different model classes making it suitable for real-world applications.

	Log. Reg. (Start/Avg)	2 layer NN (Start/Avg)	4 layer NN (Start/Avg)	6 layer NN (Start/Avg)
Elec	59.5/57.5	78.6/76.1	76.4/76.1	78.3/76.2
Weat	72.5/69.7	75.1/72.0	72.0/73.6	71.1/74.4

Table 7: Avg classification accuracy of TRIPODD for different model class. *Start/Avg* implies the start and final average accuracies of the base model on the respective datasets.

6. Related Work

Distribution shifts over time and its detection has been studied extensively in literature under different names (Tsymbal, 2004; Gama et al., 2014; Lu et al., 2018; Sato et al., 2021; Gonçalves Jr et al., 2014; Patil et al., 2021; Agrahari & Singh, 2021). The most popular among them is *concept drift* (Gama et al., 2014; Tsymbal, 2004). 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*) (Gama et al., 2014), or both. *Dataset shift* (Moreno-Torres et al., 2012) 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. This looks similar to real concept drift, but is not, as shown in Fig 1. Besides, in practice, we may not have access to the underlying distributions or have large number of data samples to accurately learn the distribution, but only have access to a model along with a few samples, which makes our work relevant and contemporary.

Methods such as LSDD-CDT (Bu et al., 2016), Marginal (Žliobaite, 2010) and Conditional Test (Kulinski et al., 2020) study covariate shift. Focusing on the covariate distribution can make these methods vulnerable to benign drifts in real-world data streams. Hence, these methods are known to produce a lot of false positives on real-world datasets, thus limiting their usefulness in practice. On the other hand, posterior shift methods (also called real concept drift methods) such as MDDM (Pesaranghader et al., 2018), DDM (Gama et al., 2004), EDDM (Baena-Garcıa et al., 2006) and ADWIN (Bifet & Gavalda, 2007) track the misclassifications of a classifier and detect drift when the distribution of error changes, i.e. track change in the posterior distribution by using cues from a given model. Such methods are generally robust to benign drifts in data streams and hence are more popular in practice. We study model drift instead, which can be more contemporarily practical. Hence, our method does not fall into the above categories as there may be no model drift in posterior or covariate shift (Fig 1).

From an interpretability perspective, while most popular drift detection methods including MDDM (Pesaranghader et al., 2018), DDM (Gama et al., 2004), EDDM (Baena-Garcıa et al., 2006), ADWIN (Bifet & Gavalda, 2007) and KSWIN (Raab et al., 2020) 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., Webb et al (Webb et al., 2018) 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. Pratt et al (Pratt & Tschapek, 2003) developed a visualization tool that used parallel histograms to study concept drift, that a user could visually inspect. *Feature-interpretable methods* (Žliobaite, 2010; Kulinski et al., 2020; Demšar & Bosnić, 2018; Lobo et al., 2021) attempt to detect drift and simultaneously notify which features might be responsible for it. Marginal (Žliobaite, 2010) performs a feature-wise KS test, while Conditional Test (Kulinski et al., 2020) performs a hypothesis test to check for change in distribution of each feature conditioned on rest of input features. (Kulinski et al., 2020) addresses the adversarial drift detection problem in their work which is slightly different from the standard drift detection problem. Lobo et al (Lobo et al., 2021) 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, which focus on concept drift or covariate shift. Using model risk allows us to apply our method in all kinds of prediction tasks, viz. classification and regression, for instance. A relatively older method (Harel et al., 2014) also uses a notion of model risk to study concept drift, but does not address interpretability. Our feature-sensitive model drift definition (Sec 3) allows us to model drift in an interpretable manner, as well as provide theoretical guarantees that are desirable for a hypothesis testing framework.

7. Conclusions

In this paper, we studied a contemporarily relevant, albeit less-studied, problem of model drift detection. Importantly, we provide the first interpretable method towards this objective. We formulated a definition for model drift in terms of input features and used it via a feature-wise hypothesis testing framework for detecting model drift in an interpretable manner. TRIPODD detects model drift and localizes the important features using model, and hence does not need white-box access to the model. This can be desirable in areas like finance and healthcare with privacy considerations. The test statistic used in TRIPODD enjoys guarantees on on test power. As TRIPODD uses only model risk, it can be applied to both classification and regression tasks, thus making it more generic than existing interpretable methods. We conduct experiments on 4 synthetic and 4 real-world datasets to study the performance of TRIPODD w.r.t. baseline methods and observe that it performs favorably w.r.t. black-box state-of-the-art methods and performs better than interpretable methods on real-world datasets. Our interpretability study showed that our method was able to focus on relevant features and gave sparse interpretations.

References

Agrahari, S. and Singh, A. K. Concept drift detection in data stream mining: A literature review. *Journal of King Saud University-Computer and Information Sciences*, 2021. 8

Baena-Garcıa, M., del Campo-Ávila, J., Fidalgo, R., Bifet, A., Gavalda, R., and Morales-Bueno, R. Early drift detection method. In *Fourth international workshop on knowledge discovery from data streams*, volume 6, pp. 77–86, 2006. 2, 8, 9

Bifet, A. and Gavalda, R. Learning from time-changing data with adaptive windowing. In *Proceedings of the 2007 SIAM international conference on data mining*, pp. 443–448. SIAM, 2007. 2, 6, 8, 9

Bifet, A., Holmes, G., Pfahringer, B., Kranen, P., Kremer, H., Jansen, T., and Seidl, T. Moa: Massive online analysis, a framework for stream classification and clustering. In *Proceedings of the first workshop on applications of pattern analysis*, pp. 44–50. PMLR, 2010. 5, 16

Bu, L., Alippi, C., and Zhao, D. A pdf-free change detection test based on density difference estimation. *IEEE transactions on neural networks and learning systems*, 29 (2):324–334, 2016. 8

Castro, J., Gómez, D., and Tejada, J. Polynomial calculation of the shapley value based on sampling. *Computers & Operations Research*, 36(5):1726–1730, 2009. 17

Catav, A., Fu, B., Zoabi, Y., Meilik, A. L. W., Shomron, N., Ernst, J., Sankararaman, S., and Gilad-Bachrach, R. Marginal contribution feature importance-an axiomatic approach for explaining data. In *International Conference on Machine Learning*, pp. 1324–1335. PMLR, 2021. 3, 4, 7, 17, 18

Chang, S., Zhang, Y., Yu, M., and Jaakkola, T. Invariant rationalization. In *International Conference on Machine Learning*, pp. 1448–1458. PMLR, 2020. 6

Costa, J., Silva, C., Antunes, M., and Ribeiro, B. Concept drift awareness in twitter streams. In *International Conference on Machine Learning and Applications*, pp. 294–299. IEEE, 2014. 2

Covert, I., Lundberg, S. M., and Lee, S.-I. Understanding global feature contributions with additive importance measures. *Advances in Neural Information Processing Systems*, 33:17212–17223, 2020. 17

Crabbé, J. and Van Der Schaar, M. Explaining time series predictions with dynamic masks. In *International Conference on Machine Learning*, pp. 2166–2177. PMLR, 2021. 6

Davis, S. E., Lasko, T. A., Chen, G., and Matheny, M. E. Calibration drift among regression and machine learning models for hospital mortality. In *AMIA Annual Symposium Proceedings*, volume 2017, pp. 625. American Medical Informatics Association, 2017. 1

De Vito, S., Massera, E., Piga, M., Martinotto, L., and Di Francia, G. On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario. *Sensors and Actuators B: Chemical*, 129(2): 750–757, 2008. 5, 16

Demšar, J. and Bosnić, Z. Detecting concept drift in data streams using model explanation. *Expert Systems with Applications*, 92:546–559, 2018. 9

Efron, B. Bootstrap methods: another look at the jackknife. In *Breakthroughs in statistics*, pp. 569–593. Springer, 1992. 5

Elwell, R. and Polikar, R. Incremental learning of concept drift in nonstationary environments. *IEEE Transactions on Neural Networks*, 22(10):1517–1531, 2011. 5, 7, 16

Friedman, J. H. Multivariate adaptive regression splines. *The annals of statistics*, 19(1):1–67, 1991. 5, 6, 16

Gama, J., Medas, P., Castillo, G., and Rodrigues, P. Learning with drift detection. In *Brazilian symposium on artificial intelligence*, pp. 286–295. Springer, 2004. 2, 6, 8, 9

Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., and Bouchachia, A. A survey on concept drift adaptation. *ACM computing surveys (CSUR)*, 46(4):1–37, 2014. 2, 8

Gonçalves Jr, P. M., de Carvalho Santos, S. G., Barros, R. S., and Vieira, D. C. A comparative study on concept drift detectors. *Expert Systems with Applications*, 41(18): 8144–8156, 2014. 8

Harel, M., Mannor, S., El-Yaniv, R., and Crammer, K. Concept drift detection through resampling. In *International conference on machine learning*, pp. 1009–1017. PMLR, 2014. 9

Harries, M. and Wales, N. S. Splice-2 comparative evaluation: Electricity pricing. 1999. 5, 7, 16

Jordaney, R., Sharad, K., Dash, S. K., Wang, Z., Papini, D., Nouretdinov, I., and Cavallaro, L. Transcend: Detecting concept drift in malware classification models. In *USENIX Security Symposium (USENIX Security 17)*, pp. 625–642, 2017. 2

Katakis, I., Tsoumakas, G., and Vlahavas, I. An ensemble of classifiers for coping with recurring contexts in data streams. In *ECAI 2008*, pp. 763–764. IOS Press, 2008. 5, 16

Khamassi, I., Sayed-Mouchaweh, M., Hammami, M., and Ghédira, K. Discussion and review on evolving data streams and concept drift adapting. *Evolving systems*, 9 (1):1–23, 2018. 1

Kulinski, S., Bagchi, S., and Inouye, D. I. Feature shift detection: Localizing which features have shifted via conditional distribution tests. In *Neural Information Processing Systems*, 2020. 2, 6, 8, 9, 15

Lobo, J. L., Del Ser, J., Osaba, E., Bifet, A., and Herrera, F. Curie: a cellular automaton for concept drift detection. *Data Mining and Knowledge Discovery*, 35(6): 2655–2678, 2021. 9

Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., and Zhang, G. Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*, 31(12): 2346–2363, 2018. 5, 8

Minne, L., Eslami, S., De Keizer, N., De Jonge, E., De Rooij, S. E., and Abu-Hanna, A. 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):40–46, 2012. 1

Montiel, J., Read, J., Bifet, A., and Abdessalem, T. Scikit-multiflow: A multi-output streaming framework. *The Journal of Machine Learning Research*, 19(1):2915–2914, 2018. 6

Moons, K. G., Kengne, A. P., Grobbee, D. E., Royston, P., Vergouwe, Y., Altman, D. G., and Woodward, M. Risk prediction models: Ii. external validation, model updating, and impact assessment. *Heart*, 98(9):691–698, 2012. 1

Moreno-Torres, J. G., Raeder, T., Alaiz-Rodríguez, R., Chawla, N. V., and Herrera, F. A unifying view on dataset shift in classification. *Pattern recognition*, 45(1):521–530, 2012. 8

Müller, M. and Salathé, M. Addressing machine learning concept drift reveals declining vaccine sentiment during the covid-19 pandemic. *arXiv preprint arXiv:2012.02197*, 2020. 2

Okhrati, R. and Lipani, A. A multilinear sampling algorithm to estimate shapley values. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 7992–7999. IEEE, 2021. 17

Patil, M. A., Kumar, S., Kumar, S., and Garg, M. Concept drift detection for social media: A survey. In *2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, pp. 12–16. IEEE, 2021. 8

Pesaranghader, A., Viktor, H. L., and Paquet, E. A framework for classification in data streams using multi-strategy learning. In *International conference on discovery science*, pp. 341–355. Springer, 2016. 5, 16

Pesaranghader, A., Viktor, H. L., and Paquet, E. Mcdiarmid drift detection methods for evolving data streams. In *International Joint Conference on Neural Networks (IJCNN)*, pp. 1–9. IEEE, 2018. 2, 6, 8, 9

Pratt, K. B. and Tschapek, G. Visualizing concept drift. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 735–740, 2003. 9

Raab, C., Heusinger, M., and Schleif, F.-M. Reactive soft prototype computing for concept drift streams. *Neurocomputing*, 416:340–351, 2020. 6, 9

Sato, D. M. V., De Freitas, S. C., Barddal, J. P., and Scalabrin, E. E. A survey on concept drift in process mining. *ACM Computing Surveys (CSUR)*, 54(9):1–38, 2021. 8

Shapely, L. A value for n-person games. contributions to the theory of games, 1953. 7, 17, 18

Shapley, L. S. *A Value for N-Person Games*. RAND Corporation, Santa Monica, CA, 1952. doi: 10.7249/P0295. 3

Tahmasbi, A., Jothimurugesan, E., Tirthapura, S., and Gibbons, P. B. Driftsurf: A risk-competitive learning algorithm under concept drift. *arXiv preprint arXiv:2003.06508*, 2020. 5

Tsymbal, A. The problem of concept drift: definitions and related work. *Computer Science Department, Trinity College Dublin*, 106(2):58, 2004. 8

Vapnik, V. Principles of risk minimization for learning theory. *Advances in neural information processing systems*, 4, 1991. 2

Webb, G. I., Lee, L. K., Goethals, B., and Petitjean, F. Analyzing concept drift and shift from sample data. *Data Mining and Knowledge Discovery*, 32(5):1179–1199, 2018. 9

Zenisek, J., Holzinger, F., and Affenzeller, M. Machine learning based concept drift detection for predictive maintenance. *Computers & Industrial Engineering*, 137: 106031, 2019. 2

Žliobaite, I. Change with delayed labeling: When is it detectable? In *International Conference on Data Mining Workshops*, pp. 843–850. IEEE, 2010. 2, 6, 8, 9

Appendix: Interpretable Model Drift Detection

In this section, we include proofs and additional details that we could not include in the main paper due to space constraints. In particular, this appendix contains the following:

- More Interpretability Results and Details
- Ablation on r value
- Proof for Theorem
- Bootstrapping Procedure
- Dataset and Hyperparameter details
- Time Complexity
- Feature-sensitive Model Drift: Brief Discussion
- Broader Impact and Limitations

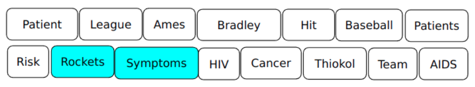
A. More Interpretability Results and Details

A.1. User Study for Results on USENET2 Dataset

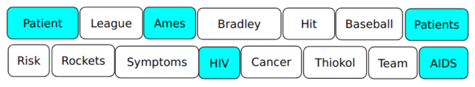
We conducted a user-study (17 participants) to provide additional context to our interpretability results, presented in Figure 2 of our main paper. We asked the study participants to choose between the outputs of three methods – TRIPODD (ours), Conditional Test and Marginal, for five questions on the USENET2 dataset (note that the participants were blind to the method used for each result while carrying out the study). The questions are as follows – (Q1) "*Which output do you think best highlights words relevant to the shift in topic from medicine to space?(Choose all that apply)*", (Q2) "*Which output has the least number of redundant words relevant to the shift in topic from medicine to space? (Choose all that apply)*", (Q3) "*Which output satisfied you for providing words responsible for the change in topic from medicine to space? (Choose all that apply)*", (Q4) "*Which output provides a sparsest selection of words responsible for shift in topic from medicine to space?(Choose the best one)*" and (Q5) "*Imagine a machine learning model for a safety-critical application where time is of the essence. As is typical, there is a shift in concept/topic, causing the model to degrade. Knowing the features(or words in this case) relevant to this shift in the topic can help a user take action. However, this knowledge should ideally be shared in a short but informative manner so that decision-making is not delayed. If you had to choose one option that gives the most useful explanation for such a safety-critical application, which one would you pick? (Choose the best one)*". Results are provided in Table 8.

Question	TRIPODD	Marg	Cond
Q1	**58.8%**	52.9%	5.9%
Q2	**93.8%**	12.5%	0.0%
Q3	**68.8%**	37.5%	12.5%
Q4	**81.3%**	12.5%	6.3%
Q5	**56.3%**	25%	18.8%

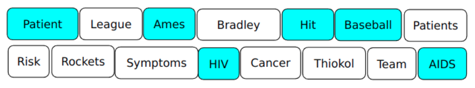
Table 8: Percentage of people who chose each method as a response to the given question. Q1, Q2, Q3 allow participants to choose more than one method, while Q4 and Q5 ask for the best method. Our explanations are preferred for all questions.



TRIPODD explanation shown in the user-study



Marginal explanation shown in the user-study



Conditional Test explanation shown in the user-study

Participants found our method to give explanations which are relevant and sparse. Furthermore, participants preferred our method's explanation for safety-critical applications – where sparse and informative explanations are required.

B. Ablation on r:

The ratio that defines the proportion of samples used for training the model to computing the test statistic in the reference window is referred to as r. In Sections 4 and 5 all the results were reported with the r value being equal to 0.8. Here we vary the value of r and study its impact on the drift detection performance of our method as measured using the proxy metric of average base model performance.

Values of $r \rightarrow$ Datasets $\downarrow$	0.6	0.7	0.8
Electricity	74.5	75.9	74.9
Weather	76.1	76.9	77.8

Table 9: Average classification accuracy for different values of r. Window size = 1000, base model = 2 layer neural network

It can be observed from Table 9 that the 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 in that case. Similarly, if r is set to a very small value(very close to 0) then the base model would have to be learnt on a small set of samples.

C. Proof for Theorem 3.3

Theorem 3.3 (Convergence of Test Power)**.** *The test power of our proposed test converges to the ideal value 1 under the alternate hypothesis. That is, suppose the alternate hypothesis H_a is true for some feature $k \in [d]$, then, for any $t > 0$,* $\lim_{n\to\infty} \mathbb{P}[\widehat{c}_n^k(h) > t] = 1$

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

Let $d^k = \Delta(S^*, k)$ and $\widehat{d}^k = \Delta_D(S^*, k)$

$$\hat{d}^k - d^k = \Delta_D\left(S^*, k\right) - \Delta\left(S^{**}, k\right) \leq \Delta_D\left(S^*, k\right) - \Delta\left(S^*, k\right) \leq \max_{S\subseteq F} |\Delta_D(S,k) - \Delta(S,k)|$$

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 that

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

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 $\hat{d}^k$.

$$|\Delta_D(S,k) - \Delta(S,k)| = |\,|\underbrace{\left(\mathcal{R}^S_{D_p}(h) - \mathcal{R}^{S\cup\{k\}}_{D_p}(h)\right)}_{e_1} - \underbrace{\left(\mathcal{R}^S_{D_q}(h) - \mathcal{R}^{S\cup\{k\}}_{D_q}(h)\right)}_{e_2}| -$$
$$|\underbrace{\left(\mathcal{R}^S_{p}(h) - \mathcal{R}^{S\cup\{k\}}_{p}(h)\right)}_{e_3} - \underbrace{\left(\mathcal{R}^S_{q}(h) - \mathcal{R}^{S\cup\{k\}}_{q}(h)\right)}_{e_4}|\,|$$
$$= |\,|e_1 - e_2| - |e_3 - e_4|\,|$$
$$\leq |e_1 - e_3 + e_4 - e_2| \leq |e_1 - e_3| + |e_4 - e_2|$$

Substituting back the values,

$$= |\left(\mathcal{R}^{S}_{D_p}(h) - \mathcal{R}^{S\cup\{k\}}_{D_p}(h)\right) - \left(\mathcal{R}^{S}_{p}(h) - \mathcal{R}^{S\cup\{k\}}_{p}(h)\right)| + |\ \left(\mathcal{R}^{S}_{q}(h) - \mathcal{R}^{S\cup\{k\}}_{q}(h)\right) - \left(\mathcal{R}^{S}_{D_q}(h) - \mathcal{R}^{S\cup\{k\}}_{D_q}(h)\right)\ |$$

$$\leq |\mathcal{R}^{S}_{D_p}(h) - \mathcal{R}^{S}_{p}(h)| + |\mathcal{R}^{S\cup\{k\}}_{p}(h) - \mathcal{R}^{S\cup\{k\}}_{D_p}(h)| + |\mathcal{R}^{S}_{D_q}(h) - \mathcal{R}^{S}_{q}(h)| + |\mathcal{R}^{S\cup\{k\}}_{q}(h) - \mathcal{R}^{S\cup\{k\}}_{D_q}(h)|$$

$$\implies |\hat{d}^k - d^k| \leq \max_{S\subseteq F}(\ |\mathcal{R}^{S}_{D_p}(h) - \mathcal{R}^{S}_{p}(h)| + |\mathcal{R}^{S\cup\{k\}}_{p}(h) - \mathcal{R}^{S\cup\{k\}}_{D_p}(h)| + |\mathcal{R}^{S}_{D_q}(h) - \mathcal{R}^{S}_{q}(h)|$$

$$+|\mathcal{R}^{S\cup\{k\}}_{q}(h) - \mathcal{R}^{S\cup\{k\}}_{D_q}(h)|\ )$$

$$= \max_{S\subseteq F}|\mathcal{R}^{S}_{D_p}(h) - \mathcal{R}^{S}_{p}(h)| + \max_{S\subseteq F}|\mathcal{R}^{S\cup\{k\}}_{p}(h) - \mathcal{R}^{S\cup\{k\}}_{D_p}(h)| + \max_{S\subseteq F}|\mathcal{R}^{S}_{D_q}(h) - \mathcal{R}^{S}_{q}(h)|$$

$$+ \max_{S\subseteq F}|\mathcal{R}^{S\cup\{k\}}_{q}(h) - \mathcal{R}^{S\cup\{k\}}_{D_q}(h)|$$

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

$$P\left(|\hat{d}^k - d^k| \leq \varepsilon\right) \leq P(\max_{S\subseteq F}|\mathcal{R}^{S}_{D_p}(h) - \mathcal{R}^{S}_{p}(h)| + \max_{S\subseteq F}|\mathcal{R}^{S\cup\{k\}}_{p}(h) - \mathcal{R}^{S\cup\{k\}}_{D_p}(h)| +$$

$$\max_{S\subseteq F}|\mathcal{R}^{S}_{D_q}(h) - \mathcal{R}^{S}_{q}(h)| + \max_{S\subseteq F}|\mathcal{R}^{S\cup\{k\}}_{q}(h) - \mathcal{R}^{S\cup\{k\}}_{D_q}(h)|\ \leq \varepsilon)$$

Using union bound of the form $P\left(\sum_{i=1}^{m} x_i \geq t\right) \leq \sum_{i=1}^{m} P\left(x_i > \frac{t}{m}\right)$ we get,

$$\begin{aligned} P\left(|\hat{d}^k - d^k| \leq \varepsilon\right) \geq 1 - P\left(\max_{S\subseteq F}|\mathcal{R}^{S}_{D_p}(h) - \mathcal{R}^{S}_{p}(h)| \geq \frac{\varepsilon}{4}\right) -\ P\left(\max_{S\subseteq F}|\mathcal{R}^{S\cup\{k\}}_{p}(h) - \mathcal{R}^{S\cup\{k\}}_{D_p}(h)| \geq \frac{\varepsilon}{4}\right) \\ - P\left(\max_{S\subseteq F}|\mathcal{R}^{S}_{D_q}(h) - \mathcal{R}^{S}_{q}(h)| \geq \frac{\varepsilon}{4}\right) -\ P\left(\max_{S\subseteq F}|\mathcal{R}^{S\cup\{k\}}_{q}(h) - \mathcal{R}^{S\cup\{k\}}_{D_q}(h)| \geq \frac{\varepsilon}{4}\right) \end{aligned} \tag{3}$$

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

$$P\left(\max_{S\subseteq F}|\mathcal{R}^{S}_{D_p}(h) - \mathcal{R}^{S}_{p}(h)| \geq \frac{\varepsilon}{4}\right) \leq P\left(\bigcup_{S\subseteq F}\left\{|\mathcal{R}^{S}_{D_p}(h) - \mathcal{R}^{S}_{p}(h)| \geq \frac{\varepsilon}{4}\right\}\right) \leq \sum_{S\subseteq F} P\left(|\mathcal{R}^{S}_{D_p}(h) - \mathcal{R}^{S}_{p}(h)| \geq \frac{\varepsilon}{4}\right)$$

$$\leq 2^{|F|+1}\exp\left(\frac{-2n\left(\frac{\varepsilon}{4}\right)^2}{(M-m)^2}\right) = 2^{|F|+1}\exp\left(\frac{-n\varepsilon^2}{8(M-m)^2}\right)$$

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

We can get a similar bound for the rest of the terms in the RHS of equation 3,

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

$$P\left(\max_{S\subseteq F}|\mathcal{R}^{S}_{D_q}(h) - \mathcal{R}^{S}_{q}(h)| \geq \frac{\varepsilon}{4}\right) \leq 2^{|F|+1}\exp\left(\frac{-n\varepsilon^2}{8(M-m)^2}\right)$$

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

Thus, we can simplify Equation 3 and also show convergence in probability of $\widehat{d^k}$ as follows:

$$P\left(|\hat{d}^k - d^k| \leq \varepsilon\right) \geq 1 - 2^{|F|+3}\exp\left(\frac{-n\varepsilon^2}{8(M-m)^2}\right) \implies \lim_{n\to\infty} P\left(|\hat{d}^k - d^k| > \varepsilon\right) = 0\ \ \forall \varepsilon > 0 \tag{4}$$

Convergence of Test Power

From Equation 4 we can infer the following:

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

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

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

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

$$\implies \mathbb{P}\left(\hat{c}_n^k \geq t\right) \geq 1 - 2^{|F|+3} \exp\left(\frac{-\left(c_n^k - t\right)^2}{8n(M-m)^2}\right) = 1 - 2^{|F|+3} \exp\left(\frac{-n\left(d^k - \frac{t}{n}\right)^2}{8(M-m)^2}\right) \tag{5}$$

Under the alternate hypothesis, it is easy to show that $d^k > 0$(necessary for applying Hoeffding's inequality on ε), therefore we can conclude the following,

$$\lim_{n\to\infty} P\left(\hat{c}^k \geq t\right) = 1 \ \forall t > 0$$

□

D. Bootstrapping Procedure

In section 3.2 we discussed the algorithm of TRIPODD where we used the bootstrap test to calculate thresholds for checking which features rejected the null hypothesis. This section consists of the algorithm we use for performing this bootstrap test. We follow Kulinski et al (Kulinski et al., 2020) in this algorithm.

Algorithm 2 BOOTSTRAP (Bootstrap With Bonferroni Correction)

```
Require: $h, \alpha, K, \mathcal{Z}_R, \mathcal{Z}_N$
0: $n, d \leftarrow$ DIM($S_R$) {$n$: Num of samples, $d$: Num of feats}
0: ScoreMatrix $\leftarrow$ ZEROS(($K, d$)) {$K \times d$ size null matrix}
0: $M \leftarrow$ MERGE($Z_R, Z_N$) {Concatenate the two sets of samples}
0: for $i$ in $[1, 2, \ldots, K]$ do {Calculate for $K$ bootstrap samples}
0:     $\widetilde{\mathcal{Z}}_R \leftarrow$ SAMPLE($M$,$n$) {Sample $n$ pts with replacement}
0:     $\widetilde{\mathcal{Z}}_N \leftarrow$ SAMPLE($M$,$n$) {Sample $n$ pts with replacement}
0:     FINETUNE($h, \widetilde{\mathcal{Z}}_R$) {Finetune for 2 epochs(Kulinski et al., 2020)}
0:     Compute $c_n^k(h), \forall k \in [d]$, using $\widetilde{\mathcal{Z}}_R, h$ in Equation 1 from the main paper.
0:     $[\text{ScoreMatrix}]_i = [c_n^1, c_n^2, \ldots, c_n^d]$
0: end for
0: return $(1 - \frac{\alpha}{d})$ quantile of $c_n^k$ values stored in ScoreMatrix $\forall k$ =0
```

To obtain the threshold to reject the null hypothesis, we perform bootstrap sampling $K = 50$ number of times. Specifically, we merge and shuffle samples from p and q distributions, and then pick K two-samples from this mixture. This simulates the null hypothesis. Now, 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). We use α = 0.05 as the significance level of our test as is standard in hypothesis testing literature. Similar to (Kulinski et al., 2020), while computing the test statistic during the bootstrap test, we refit our models for each of the K two-samples. Specifically, we train models on the originally given reference and new samples windows, and update these models (1-2 epochs of finetuning) for each bootstrap episode.

E. Dataset and Hyperparameter Details

This section describes all datasets used in our experiments and hyperparameters used in our experiments.

E.1. Datasets

This section briefly explains all the datasets used in our experiments and also highlights the relevance to concept drift in each of them.

Hyperplane (Bifet et al., 2010): A hyperplane is used here to simulate time-varying concepts. As the position and orientation of the hyperplane is changed smoothly, the old classifier (trained to separate datapoints lying on different halfspaces of the hyperplane) gets outdated. The magnitude of change is 0.001 to simulate a gradual drift. The drift locations are not known in this dataset.

Mixed (Pesaranghader et al., 2016): This dataset has four input features (x_1, x_2, x_3, x_4), where x_1 and x_2 are boolean and x_3, x_4 are uniformly distributed in [0, 1]. Label of each data is determined to be positive if two of x_1, x_2, and $x_4 < 0.5 + 0.3sin(3\pi x_3)$ conditions hold. A drift is caused when the distribution of x_3 and x_4 are perturbed i.e. changed from a uniform to a non-uniform distribution.

Friedmann (Friedman, 1991): It is a regression data set that consists of 6 features that are each drawn from a uniform distribution from the interval [0, 1]. The output variable is given by $10sin(\pi x_4 x_5) + 20(x_2 - 0.5)^2 + 10 * x_1 + 5x_3 + \mathcal{N}(0, 1)$. Drift is simulated by changing the range of input features from [0,1] to [-1,2].

Sine (Pesaranghader et al., 2016): This dataset contains two attributes (x_1, x_2), uniformly distributed in [0, 1]. The output label is determined by using the curve $x_2 = sin(x_1)$, and data points lying below it are classified as positive. The first drift is created by reversing the labels. For the next drift the curve (or labeling function) is changed. The next drift is simulated by reversing the labels.

USENET2 (Katakis et al., 2008): This dataset is created by labeling a stream of email messages as interesting or not interesting depending on the labeler's interest. They contain data for 5 time periods of 300 samples each. A drift is present after each time period because of change in user (or labeler) interest. We remove stop words such as pronouns, dates, numbers and articles from this dataset. Examples of such words from the USENET2 dataset are: *'taken', 'eng', 'wrench', '1993apr20', '22', 'young', 'bob', 'mda'*.

Electricity (Harries & Wales, 1999): This dataset contains information about the New South Wales Electricity Market in Australia. Input features correspond to price, demand and amount of energy transferred between two states in Australia. The class label indicates whether the transfer price (cost to transfer electricity from one state to another) is increased or decreased relative to a moving average of the last 24 hours. Drift is expected due to changes in consumption habits, change in electricity board and other unexpected deviations.

Weather (Elwell & Polikar, 2011): This dataset contains weather data from 1949-1999 about rain prediction. It has eight different input features relevant for predicting rainfall such as temperature, pressure, wind speed, etc. Drift is expected due to change in seasons, climate and other natural factors.

Air Quality (De Vito et al., 2008): This dataset contains measurements from five metal oxide chemical sensors and a temperature, and a humidity sensor. The prediction task is to predict the benzene concentration, which is a proxy for air pollution. A drift is expected due to seasonal changes in the weather.

E.2. Hyperparameters

This section describes various hyperparameters used in our experiments to facilitate reproducibility.

Neural network for modeling posterior distribution: Across all our experiments (including ablation studies in Section 5), we used 3 neural network architectures:

1. 2-layer Neural Network: First layer has 1024 neurons and second has 512 neurons. This was used in all our experiments reported in Section 4.

2. 4-layer Neural Network: We used 1024, 512,256,128 layers from first to fourth layer respectively. This was used in our ablation studies reported in Section 5.

3. 6-layer Neural Network: We used 1024, 512,256,128,128,64 layers from first to sixth layer respectively. This was used in our ablation studies reported in Section 5.

We used Dropout of 10% after every layer in each neural network. An $L2$ weight decay regularizer with co-efficient 0.001 in the loss term was used during training. ReLU activation function in all our networks. We use a learning rate of 1e-04, batch size of 16 and Adam optimizer for all our experiments.

Parameters used for drift detection algorithm: For TRIPODD we use $K = 50$ number of bootstraps, $\delta = 10$ for window sizes less than 1000 and $\delta = 50$ for window sizes greater than or equal to 1000.

F. Time Complexity

Following up our discussion in Section 4, we would like to state that TRIPODD is the only interpretable drift detection method that comprehensively considers feature interactions. This in turn forces a time complexity that grows with the number of input features, similar to MCI (Catav et al., 2021) and Shapley value-like feature importance scores (Shapely, 1953), which take feature interaction into account in a similar manner. Below, we report the time taken by TRIPODD and other baseline interpretable methods on two real-world datasets while operating on a window (both reference and new samples) of size 1500 and with a base model of 4-layer neural network. Please note that although Marg and Cond use lesser time than TRIPODD, they indicate drift at every time step, i.e. many false positives on these datasets, which hampers their utility in practice. As mentioned in the main paper, real-world datasets are captured over years often with sampling rates across hours; providing an interpretable drift detector that is in order of seconds is pseudo-realtime for such settings and practically relevant.

Methods $\rightarrow$ Datasets $\downarrow$	Marg	Cond	Ours
Electricity(# features = 5)	2	16	100
Weather(# features = 9)	2	18	400

Table 10: Time taken (in secs) by different interpretable drift detection methods tp detect drift for a reference and new samples window of size 1500 samples each.

For datasets with large number of features, Shapley value (and MCI)-based frameworks are known to use sampling techniques (Castro et al., 2009; Okhrati & Lipani, 2021; Covert et al., 2020) to avoid computing scores for all possible subsets of features. We leverage these developments and use the random sampling technique proposed by Covert et al.(A Covert et al., 2020) to reduce the computational overhead. Reducing the computational overhead further by using efficient approximation techniques is a potential future work direction.

G. Feature-sensitive Model Drift: Brief Discussion

Continuing from our discussion of feature-sensitive model drift in Section 2, we note that our feature-sensitive model drift definition, and subsequently our test statistic, is inspired from the MCI (Catav et al., 2021) feature importance measure. MCI is based on a set of axioms which ensures that presence of correlation between features in a dataset does not negatively affect their ranking w.r.t. output prediction. Thus, it improves upon Shapley values when handling correlated features in real-world datasets. Furthermore, MCI also satisfies nice properties similar to Shapley values (slightly relaxed), making it suitable for measuring feature importance.

H. Broader Impact and Limitations

Positive Social Impact: Interpretable methods instill trust in machine learning models as they provide users with some insights into their decision making, thus our method could help practitioners to get alerts on model degradation and also get help to get insights about the same.

Negative Social Impact: Our method's performance is dependent on the base model and the model class it belongs to. So, the interpretations given by our method are model-dependent and hence should not be used to infer about the underlying data distribution directly. We note however that the growing use of machine learning models make a model-dependent drift detector of practical value by itself. We use feature-sensitive model drift (Defn 2) for interpretable model drift detection where we check each feature's interaction with every possible combination of other features. This can require a large number of computations, similar to computing Shapley Values (Shapely, 1953) or MCI (Catav et al., 2021)-like scores. We however address this using sampling-based techniques to reduce the number of computations required for datasets with high number of features.