Machine Learning (2023) 112:4863–4903
https://doi.org/10.1007/s10994-023-06385-y

# Incremental permutation feature importance (iPFI): towards online explanations on data streams

Fabian Fumagalli[1] · Maximilian Muschalik[2] · Eyke Hüllermeier[2] · Barbara Hammer[1]

Received: 5 June 2023 / Revised: 5 June 2023 / Accepted: 12 July 2023 /
Published online: 20 September 2023

## Abstract

Explainable artificial intelligence has mainly focused on static learning scenarios so far. We are interested in dynamic scenarios where data is sampled progressively, and learning is done in an incremental rather than a batch mode. We seek efficient incremental algorithms for computing feature importance (FI). Permutation feature importance (PFI) is a well-established model-agnostic measure to obtain global FI based on feature marginalization of absent features. We propose an efficient, model-agnostic algorithm called iPFI to estimate this measure incrementally and under dynamic modeling conditions including concept drift. We prove theoretical guarantees on the approximation quality in terms of expectation and variance. To validate our theoretical findings and the efficacy of our approaches in incremental scenarios dealing with streaming data rather than traditional batch settings, we conduct multiple experimental studies on benchmark data with and without concept drift.

**Keywords** Incremental learning · Explainable artificial intelligence · Feature importance · Permutation feature importance

Editors: Fabio Vitale, Tania Cerquitelli, Marcello Restelli, Charalampos Tsourakakis.

Fabian Fumagalli and Maximilian Muschalik have contributed equally to this work.

✉ Fabian Fumagalli
ffumagalli@techfak.uni-bielefeld.de

✉ Maximilian Muschalik
Maximilian.Muschalik@ifi.lmu.de

Eyke Hüllermeier
eyke@lmu.de

Barbara Hammer
bhammer@techfak.uni-bielefeld.de

[1] Bielefeld University, 33594 Bielefeld, Germany

[2] LMU Munich, 80539 Munich, Germany

## 1 Introduction

Online learning from dynamic data streams is a prevalent machine learning (ML) approach for various application domains (Bahri et al., 2021). For instance, predicting energy consumption for individual households can foster energy-saving strategies such as load-shifting. Concept drift resulting from environmental changes, such as pandemic-induced lock-downs, drastically impacts the energy consumption patterns necessitating online ML (García-Martín et al., 2019). Explaining these predictions yields a greater understanding of an individual's energy use and enables prescriptive modeling for further energy-saving measures (Wastensteiner et al., 2021). For black-box ML methods, so-called post-hoc XAI methods seek to explain single predictions or entire models in terms of the contribution of specific features (Adadi & Berrada, 2018).

We are interested in feature importance (FI) as a global assessment of features, which indicates their respective relevance to the given task and model. A prominent representative of model-agnostic FI measures is the permutation feature importance (PFI), which, in its original form, has been introduced for tree-based models in Breiman (2001) with various applications and extensions (Strobl et al., 2007, 2008; Altmann et al., 2010; Hapfelmeier et al., 2014; Zhu et al., 2015; Gregorutti et al., 2015, 2017). Recent work (Fisher et al., 2019) adapts PFI to a model-agnostic FI measure (model reliance) and establishes important theoretical guarantees. Albeit its limitations (Hooker et al., 2019; Fisher et al., 2019), we focus on PFI as a well-established, efficiently implementable, model-agnostic FI measure, which has served as a baseline for various more powerful extensions (Casalicchio et al., 2018; Covert et al., 2020; Molnar et al., 2020; König et al., 2021). So far, PFI requires a holistic view of the entire dataset in a static batch learning environment, which does not account for changes in the model structure, efficient anytime computations or sparse storage capabilities in data stream settings.

More generally, explainable artificial intelligence (XAI) has been studied mainly in the batch setting, where learning algorithms operate on static datasets. In scenarios where data does not fit into memory or computation time is strictly limited, like in progressive data science for big datasets (Turkay et al., 2018), or rapid online learning from data streams (Bahri et al., 2021), high computation times prohibit the use of traditional FI or XAI measures. Incremental, time- and memory-efficient implementations that provide anytime results have received much attention in recent years (Losing et al., 2018; Montiel et al., 2020). In particular, incremental algorithms enable a lifelong adaptation of machine learning technologies and their applications to possibly infinite data streams, addressing computational challenges as well as the challenge of dealing with changes of the underlying data distribution called drift. In this article, we are interested in efficient incremental algorithms for FI (see Fig. 1). Especially in the context of drifting data distributions, this task is particularly relevant—but also challenging, as many common FI methods are already computationally costly in the batch setting. While we focus with our implementation and theoretical results on PFI, our methodology of incremental FI could also be extended to other XAI measures.

*Contribution*

We propose an incremental variant of PFI as a model-agnostic global FI estimator which is capable of dealing with data streams of arbitrary length in limited memory and linear time. Our algorithm can be applied to any model that is incrementally learned on a data stream and provides anytime explanations that immediately react to changes in the model and the underlying data distribution in case of concept drift. The main idea is that these

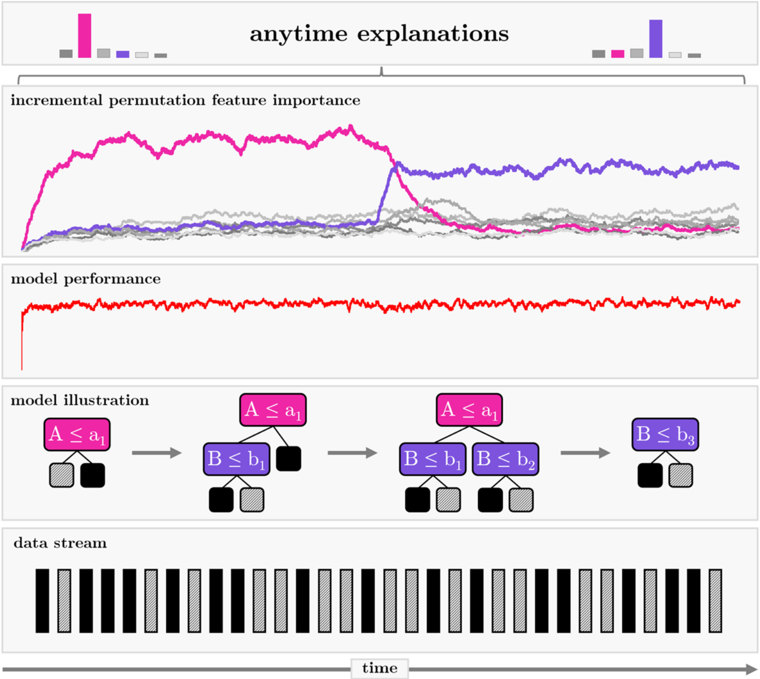


**Fig. 1** Incremental feature importance on an electricity data stream to create anytime explanations. Concept drift in the data (rectangles) lead to model adaption without visible changes in the model's performance

estimates are efficiently updated at each time step by computing a one-sample estimate of FI, which is then exponentially averaged over time. To approximate marginal feature distributions we introduce two sampling strategies, which can be applied in scenarios with and without concept drift. The core idea, inspired by reservoir sampling (Vitter, 1985), is to efficiently maintain a reservoir to sample observations that are used to approximate the marginal distribution. Our core contributions include:

- We introduce iPFI as an *incremental* and *model-agnostic* estimator for *global* FI by constructing an online variant of PFI (Sect. 3). Up to our knowledge, this constitutes the first mathematically substantiated approach for online global FI estimation. In contrast to PFI, iPFI reacts to concept drift in non-stationary environments and provides an explanation stream alongside the data stream in linear time and constant memory. The explanation stream can be utilized for further downstream tasks, such as inspecting possible causes for observed drift.
- We motivate iPFI by establishing the concrete connection of permutation tests (Definition 2) (Breiman, 2001) and model reliance (Definition 3) (Fisher et al., 2019) in the batch setting (Theorem 1). This finding extends on (Fisher et al., 2019, Appendix A.3) and shows only properly scaled permutation tests are unbiased estimates of global FI.
- We provide two sampling strategies for iPFI to incrementally compute marginal feature distributions and establish theoretical guarantees regarding bias, variance, and approximation error in terms of a single sensitivity parameter in a static and dynamic learning scenario.

- We implement iPFI and conduct experiments on its ability to efficiently provide anytime global FI values under different types of concept drift, as well as its approximation quality compared to batch permutation tests in static modeling scenarios.

All experiments and algorithms are publicly available and integrated natively into the well-known incremental learning framework *river* (Montiel et al., 2020).[1]

*Related work* A variety of model-agnostic local FI methods (Ribeiro et al., 2016; Lundberg & Lee, 2017; Lundberg et al., 2020; Covert & Lee, 2021) exist that provide relevance values for single instances. In addition, model-specific variants have been proposed for neural networks (Bach et al., 2015; Selvaraju et al., 2017) and trees (Lundberg et al., 2020). In contrast, global FI methods provide relevance values across all instances. PFI (or permutation tests) (Breiman, 2001) are a prominent global FI approach that has been widely applied (Archer & Kimes, 2008; Calle & Urrea, 2011; Wang et al., 2016), studied and extended (Strobl et al., 2007, 2008; Altmann et al., 2010; Hapfelmeier et al., 2014; Zhu et al., 2015; Gregorutti et al., 2015, 2017) for tree-based models. The method has recently been introduced as a model-agnostic approach in Fisher et al. (2019) and extended to scenarios with strongly correlated features in Molnar et al. (2020); König et al. (2021). In this regard, our definition of global FI also relates to an ongoing debate, if absent features should be marginalized using the conditional distribution (Aas et al., 2021; Frye et al., 2021) or the marginal distribution (Janzing et al., 2020), as proposed by PFI, where it was argued that the choice should depend on the application (Chen et al., 2020), and the marginal distribution was used as an approximation of the conditional distribution (Covert et al., 2020; Lundberg & Lee, 2017). A particular popular extension is SAGE, a Shapley-based (Shapley, 1953) approach, which averages marginal feature contributions over arbitrary subsets of marginalized features. It has been proposed and compared with existing methods in Covert et al. (2020) and a closely related idea was previously introduced in Casalicchio et al. (2018), called SFIMP. As calculating FI values is computationally expensive, especially for Shapley-based methods, more efficient approaches such as FastSHAP (Jethani et al., 2021) have been introduced. Yet, none of the above methods and extensions natively support an incremental or dynamic setting in which the underlying model and its global FI can rapidly change due to concept drift.

An initial approach to explaining model changes by computing differences in FI utilizing drift detection methods is Muschalik et al. (2022). However, this does not constitute an incremental FI measure. The explanations are created with a time delay and without efficient anytime calculations. A first step towards anytime FI values has been proposed for online random forests by computing mean decrease in impurity (MDI) and accuracy (MDA) over time by using online confusion matrices (Cassidy & Deviney, 2014) or maintaining node statistics incrementally (Gomes et al., 2019). While online feature scores are of particular interest in streaming scenarios, these methods are limited to a specific model class, need access to the inherent model structure and cannot be extended to a model-agnostic approach. They further do not provide any theoretical guarantees about the approximation quality in comparison to the batch versions.

Similar to batch learning, incremental FI values could be used in further downstream tasks. As an example, incremental FI is also relevant to the field of incremental feature selection, where FI is calculated periodically with a sliding window to retain features for

[1] We provide iPFI as an open-source implementation in the *iXAI* online explanation framework available at https://github.com/mmschlk/iXAI.

the incrementally fitted model (Barddal et al., 2019; Yuan et al., 2018). Lastly, changes in FI can also be used for concept drift detection, as described in Haug et al. (2022).

In this work, we provide a mathematically substantiated model-agnostic incremental FI measure, whose time sensitivity can be controlled by a single smoothing parameter. To our knowledge, this is the first approach that combines online ML and model-agnostic XAI measures and provides extensive theoretical guarantees on its approximation quality.

## 2 Global feature importance

We consider a supervised learning scenario, where $\mathcal{X}$ is the feature space and $\mathcal{Y}$ the target space, e.g., $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{Y} = \mathbb{R}$ (regression), $\mathcal{Y} = \{0, 1\}$ (binary classification) or $\mathcal{Y} = \{0, 1\}^c$ (multiclass classification). Let $h : \mathcal{X} \to \mathcal{Y}$ be a model, which is learned from a set or stream of observed data points $z = (x, y) \in \mathcal{X} \times \mathcal{Y}$. Let $D = \{1, \ldots, d\}$ be the set of feature indices for the vector-wise feature representations of $x = (x^{(i)} : i \in D) \in \mathcal{X}$. Consider a subset $S \subset D$ and its complement $\bar{S} := D \backslash S$, which partitions the features, and denote $x^{(S)} := (x^{(i)} : i \in S)$ as the feature subset of $S$ for a sample $x$. We write $h(x^{(\bar{S})}, x^{(S)}) := h(x)$ to distinguish between features from $\bar{S}$ and $S$. For the basic setting, we assume that $N$ observations are drawn independently and identically distributed (iid) from the joint distribution of unknown random variables $(X, Y)$ and denote by $\mathbb{P}_S$ the marginal distribution of the features in $S$, i.e., $z_n := (x_n, y_n)$ from $Z_n := (X_n, Y_n) \overset{iid}{\sim} \mathbb{P}_{(X,Y)}$ and $x_n^{(S)}$ from $X_n^{(S)} \overset{iid}{\sim} \mathbb{P}_S$ for samples $n = 1, \ldots, N$.

*Feature importance* refers to the relevance of a set of features $S$ for a model $h$. To quantify FI, the key idea of measures such as PFI is to compare the model's performance when using only features in $\bar{S}$ with the performance when using all features in $D = S \cup \bar{S}$. The idea is that the "removal" of an important feature (i.e., the feature is not provided to a model) substantially decreases a model's performance. The model performance or risk is measured based on a norm $\| \cdot \| : \mathcal{Y} \to \mathbb{R}$ on $\mathcal{Y}$, e.g., the Euclidean norm, as $\mathbb{E}_{(X,Y)}[\ \|h(X) - Y\|\ ]$.

As the model is trained on all features and retraining is computationally expensive, a common method to restrict $h$ to $\bar{S}$ is to marginalize $h$ over the features in $S$. We denote the marginalized risk

$$f_S\big(x^{(\bar{S})}, y\big) := \mathbb{E}_{\tilde{X} \sim \mathbb{P}_S}\Big[\|h(x^{(\bar{S})}, \tilde{X}) - y\|\Big]. \tag{1}$$

We then define FI for a model $h$ and a feature set $S$ as the difference of the marginalized risk and the inherent risk.

**Definition 1** (*Global FI*) For a model $h$ and a subset $S \subset D$, the *global feature importance* (global FI) is defined as

$$\phi^{(S)}(h) := \underbrace{\mathbb{E}_{(X,Y)}\Big[f_S(X^{(\bar{S})}, Y)\Big]}_{\text{marginalized risk over } \mathbb{P}_S} - \underbrace{\mathbb{E}_{(X,Y)}\Big[\|h(X) - Y\|\Big]}_{\text{risk}}.$$

This global FI measures the *increase in risk* when the features in $S$ are marginalized.

***Remark 1*** Our definition is best suited and inherent for PFI (Breiman, 2001; Fisher et al., 2019) with single feature subsets. However, it is also related to a more general definition of FI given in Covert et al. (2020). Therein, FI is based on the *reduction in risk*, if features in $S$ are included compared to marginalizing all features. In contrast to our definition, it relies on the conditional distribution $X^{(\bar{S})} \mid X^{(S)}$. In practice, however, the marginal distribution is often used to approximate the conditional distribution, where both coincide, if feature independence is assumed (Lundberg & Lee, 2017; Covert et al., 2020, 2021). In this case, it directly corresponds to our definition with different notation. In the literature, it was argued that the choice of distribution should depend on the application (Chen et al., 2020). The conditional distribution was preferred in Aas et al. (2021); Frye et al. (2021), which includes causal relationships in the explanation (Chen et al., 2020), whereas the marginal distribution was preferred in Janzing et al. (2020), which explains the model independent of the relationships between the features (Janzing et al., 2020; Chen et al., 2020).

## 2.1 Empirical estimation of global FI

Given observations $(x_1, y_1), \dots, (x_N, y_N)$, we estimate global FI for a given model $h$ with the canonical estimator

$$\hat{\phi}^{(S)}_{\varphi} := \frac{1}{N}\sum_{n=1}^{N} \hat{\lambda}^{(S)}(x_n, x_{\varphi(n)}, y_n), \tag{2}$$

where $\varphi : \{1, \dots, N\} \to \{1, \dots, N\}$ represents the realization of a (possibly random) sampling strategy that chooses for $x_n$ an observation $x_{\varphi(n)}$ as a replacement value with

$$\hat{\lambda}^{(S)}(x_n, x_m, y_n) := \|h(x_n^{(\bar{S})}, x_m^{(S)}) - y_n\| - \|h(x_n) - y_n\|.$$

Given the iid assumption, it is clear that due to $X_n \perp X_{n'}$ for $n \neq n'$, the estimator is an unbiased estimator of the global FI $\phi^{(S)}(h)$, if $\varphi(n) \neq n$ for all $n = 1, \dots, N$. In the case of $\varphi(n) = n$, the term in the sum is zero as well as its expectation, which implies $\mathbb{E}[\hat{\phi}^{(S)}_{\varphi}] \leq \phi^{(S)}(h)$ for any $\varphi$. We will now discuss a well understood choice of feature subsets $S \subset D$, sampling strategy $\varphi$ and two estimators for $\phi^{(S)}(h)$.

## 2.2 Permutation feature importance (PFI)

A popular example of global FI is the well-known PFI (Breiman, 2001) that measures the importance of each feature $j \in D$ by using a set $S_j := \{j\}$. More precisely, the FI for each feature $j \in D$ is given by $\phi^{(S_j)}$ with sets $S_j$ and their complement $\bar{S}_j := D \setminus \{j\}$. The sampling strategy $\varphi$ used in PFI samples uniformly generated permutations $\varphi \in \mathfrak{S}_N$ over the set $\{1, \dots, N\}$, where each permutation has a probability of $1/N!$.

### 2.2.1 Empirical PFI

Permutation tests, as proposed initially in Breiman (2001), effectively approximate $\mathbb{E}_{\varphi}[\hat{\phi}^{(S_j)}_{\varphi}]$ by averaging over $M$ uniformly sampled random permutations. We now introduce a corrected version of the originally proposed estimator, which we refer to as *PFI* by introducing a normalizing factor $\frac{N}{N-1}$.

**Definition 2** (*PFI*) Given samples $(x_1, y_1), \dots, (x_N, y_N)$ and uniformly sampled permutations $\varphi_1, \dots, \varphi_m \overset{iid}{\sim} \text{unif}(\mathfrak{S}_N)$, we define the *PFI* estimator as

$$\textbf{PFI} : \hat{\phi}^{(S_j)} := \frac{N}{N-1} \underbrace{\frac{1}{M} \sum_{m=1}^{M} \hat{\phi}^{(S_j)}_{\varphi_m}}_{\approx \mathbb{E}_\varphi[\hat{\phi}^{(S_j)}_\varphi]}. \tag{3}$$

As discussed above, the estimator $\hat{\phi}^{(S_j)}_\varphi$ for a given $\varphi$ is an unbiased estimator for global FI $\phi^{(S_j)}(h)$, if the permutation is a derangement (no fixed points). Our version differs by the factor $\frac{N}{N-1}$ from the initially proposed approach (Breiman, 2001; Fisher et al., 2019). In the following, we show that, if the expectation over uniformly sampled permutations $\varphi \sim \text{unif}(\mathfrak{S}_N)$ is taken, our definition is an unbiased estimator of global FI. This expectation directly links PFI to model reliance (Fisher et al., 2019), which we thus refer to as *expected PFI*. While our definition of PFI is closely related to the original method (Breiman, 2001), the link to expected PFI allows to provide further theoretical results. We utilize this link in an incremental learning setting to provide theoretical guarantees.

### 2.2.2 Expected PFI

The PFI estimator can be efficiently computed but highly depends on the sampled permutations complicating the theoretical analysis. Another definition of PFI (model reliance), which was given and extensively studied in Fisher et al. (2019), is independent of sampled permutations. We refer to it as the *expected PFI*.

**Definition 3** (*Expected PFI*) Given observations $(x_1, y_1), \dots, (x_N, y_N)$ the *expected PFI* is defined as

$$\bar{\phi}^{(S_j)} := \underbrace{\frac{1}{N(N-1)} \sum_{n=1}^{N} \sum_{m \neq n} \| h(x_n^{(\bar{S}_j)}, x_m^{(S_j)}) - y_n \|}_{=: \hat{e}_{\text{switch}}} - \underbrace{\frac{1}{N} \sum_{n=1}^{N} \| h(x_n) - y_n \|}_{=: \hat{e}_{\text{orig}}}$$

The expected PFI computes the difference of the error of the model averaged over all feature instantiations $\hat{e}_{\text{switch}}$ with the model error $\hat{e}_{\text{orig}}$[2]. We now show that the expected PFI is actually the expectation over the sampling procedure $\varphi$ of PFI, which directly links Definition 2 and Definition 3. As expected PFI is an unbiased estimator for global FI, we conclude that PFI is an unbiased estimator, if properly scaled as in Definition 2.

[2] As compared to Fisher et al. (2019), we consider the loss function $L(f, (y, x_n, x_m)) := \| h(x_n^{(\bar{S}_j)}, x_m^{(S_j)}) - y \|$ and denote $\bar{\phi}^{(S_j)} := \widehat{MR}_{\text{difference}}(h)$ in our case.

**Theorem 1** *The expected PFI (model reliance) can be rewritten as a normalized expectation over uniformly sampled permutations*

$$\bar{\phi}^{(S_j)} = \frac{N}{N-1} \mathbb{E}_{\varphi \sim \mathrm{unif}(\mathfrak{S}_N)} \left[ \hat{\phi}_{\varphi}^{(S_j)} \right] \approx \hat{\phi}^{(S_j)} \tag{4}$$

*i.e. expected PFI is canonically estimated by the PFI estimator and in particular* $\bar{\phi}^{(S_j)} = \mathbb{E}_{\varphi}[\hat{\phi}^{(S_j)}]$.

Due to space restrictions, all proofs are deferred to the supplementary material in Sect. A. Theorem 1 shows that the PFI estimator $\hat{\phi}^{(S_j)}$ is a canonical Monte-Carlo estimate of the theoretically well understood expected PFI estimator $\bar{\phi}^{(S_j)}$. Both $\hat{e}_{\text{switch}}$ and $\hat{e}_{\text{orig}}$ as well as the estimator $\bar{\phi}^{(S_j)}$ are U-statistics, which implies unbiasedness, asymptotic normality and finite sample boundaries under weak conditions (Fisher et al., 2019). The variance can, thus, be directly computed and it is easy to show that $\mathbb{V}[\bar{\phi}^{(S_j)}] = \mathcal{O}(1/N)$, which by Chebyshev's inequality implies a bound on the approximation error as $\mathbb{P}(|\bar{\phi}^{(S_j)} - \phi^{(S_j)}(h)| > \epsilon) = \mathcal{O}(1/N)$. Hence, the approximation error of the expected PFI is directly controlled by the number of observations $N$ used for computation. A possible link between permutation tests and the U-statistic $\bar{\phi}^{(S_j)}$ was already suggested in (Fisher et al., 2019, Appendix A.3), where it was shown that the sum over permutations without fixed points is proportional to $\hat{e}_{\text{switch}}$. Theorem 1 shows that both approaches are directly linked, if permutation tests are properly scaled (Definition 2). The biased estimator $\frac{1}{M} \sum_{m=1}^{M} \hat{\phi}_{\varphi_m}^{(S_j)}$ appears in Breiman (2001); Fisher et al. (2019); Gregorutti et al. (2017). To our knowledge, the unbiased version in Definition 2 has not yet been introduced. In practice, while this factor does not change the relative importance scores, it should be considered when comparing PFI estimates with varying $N$. Furthermore, Theorem 1 justifies to average over repeatedly sampled realizations of $\varphi$ in order to approximate the computationally prohibitive estimator $\bar{\phi}^{(S_j)}$. In the following, we will pick up this notion when constructing an incremental FI estimator.

## 3 Incremental permutation feature importance

In incremental learning, one deals with an a priory unlimited stream of training data. The challenge is to infer a model at any time point $t$ based on the previous model and the currently observed data point, thereby using a fixed, limited amount of memory and efficient update schemes for the model. While incremental classification and regression models have been proposed (Bahri et al., 2021; Losing et al., 2018), technologies which accompany such methods by incremental explanation technologies are rare. In the following, we introduce an efficient incremental scheme for the popular PFI supported by theoretical guarantees using the link to expected PFI (model reliance) (Fisher et al., 2019).

We now consider a sequence of models $(h_t)_{t \in \mathbb{N}}$ from an incremental learning algorithm. At time $t$ the observed data is $\{(x_0, y_0), \dots, (x_t, y_t)\}$. The model is incrementally learned over time, such that at time $t$ the observation $(x_t, y_t)$ is used to update $h_t$ to $h_{t+1}$. Our goal is to efficiently provide an estimate of PFI at each time step $t$ for each feature $j \in D$ using subsets $S_j := \{j\}$. Note that our results can immediately be extended to arbitrary feature subsets $S \subset D$.

In the following, we construct an efficient incremental estimator for PFI. We first discuss how (2) can be efficiently approximated in the incremental learning scenario, given a sampling strategy $\varphi_t$. In the sequel, we will rely on a random sampling strategy which is specifically suitable for the incremental setting and easier to implement than permutation-based approaches. Note that a permutation-based approach at time $t$ is difficult to replicate in the incremental setting, as at time $s < t$ not all samples until time $t$ are available. Moreover, as the model changes over time, naively computing (2) at each time step $t$ using $N$ previous observations results in $N$ model evaluations per time step. Instead, we propose to use an estimator that averages the terms in (2) over time rather than over multiple data points at one time step. That means, we evaluate the current model only twice to compute the time-dependent quantity

$$\hat{\lambda}_t^{(S_j)}(x_t, x_{\varphi_t}, y_t) := \|h_t(x_t^{(\bar{S}_j)}, x_{\varphi_t}^{(S_j)}) - y_t\| - \|h_t(x_t) - y_t\|,$$

where $\varphi_t$ is a stochastic sampling strategy to select a previous observation with values in $\{0, \dots, t-1\}$, which we discuss in a second step in Sect. 3.1. We propose to average these calculations over time (rather than iterations over multiple data points) by using exponential smoothing. This yields to the definition of the incremental PFI (iPFI) estimator.

**Definition 4** (*iPFI*) For a data stream at time $t$ with previous observations $(x_0, y_0), \dots, (x_t, y_t)$ and a sampling strategy $(\varphi_s)_{s=t_0,\dots,t}$ for $t_0 > 0$ the *incremental PFI* (iPFI) estimator is recursively defined as

$$\textbf{iPFI} : \hat{\phi}_t^{(S_j)} := (1-\alpha)\hat{\phi}_{t-1}^{(S_j)} + \alpha\hat{\lambda}_t^{(S_j)}(x_t, x_{\varphi_t}, y_t),$$

for $t > t_0$, $\hat{\phi}_{t_0-1}^{(S_j)} := 0$, and $\alpha \in (0, 1)$.

**Algorithm 1** iPFI explanation at time $t$ for feature $j$

```
Require: : $\alpha \in (0,1)$, sampling strategy $\varphi_t$, and $\hat{\phi}_{t-1}^{(S_j)}$, current model $h_t$, and
    current observation $(x_t, y_t)$
procedure EXPLAINONE($h_t, x_t, y_t, j$)
    $x_s \leftarrow \text{Sample}(\varphi_t)$
    $\hat{\lambda}_t^{(S_j)} \leftarrow \|h_t(x_t^{(\bar{S}_j)}, x_s^{(S_j)}) - y_t\| - \|h_t(x_t) - y_t\|$
    $\hat{\phi}_t^{(S_j)} \leftarrow (1-\alpha) \cdot \hat{\phi}_{t-1}^{(S_j)} + \alpha \cdot \hat{\lambda}_t^{(S_j)}$
    $\varphi_{t+1} \leftarrow \text{UpdateSampler}(\varphi_t, x_t)$
end procedure
```

The parameter $\alpha$ is a hyperparameter that should be chosen based on the application. Note that a specific choice of $\alpha$ corresponds to a window size $N$, where $\alpha = \frac{2}{N+1}$ based on the well-known conversion formula, see e.g. (Nahmias & Olsen, 2015, p.73). Given a realization $\varphi_s$, observations $z_s := (x_s, y_s)$ from iid $Z_s := (X_s, Y_s) \overset{iid}{\sim} \mathbb{P}_{(X,Y)}$ and $x_s^{(S_j)}$ from $X_s^{(S_j)} \overset{iid}{\sim} \mathbb{P}_{S_j}$, each $\hat{\lambda}_s^{(S_j)}$ is an unbiased estimate of $\phi^{(S_j)}(h_s)$. We further require $\varphi_s \perp (X, Y)$ and denote

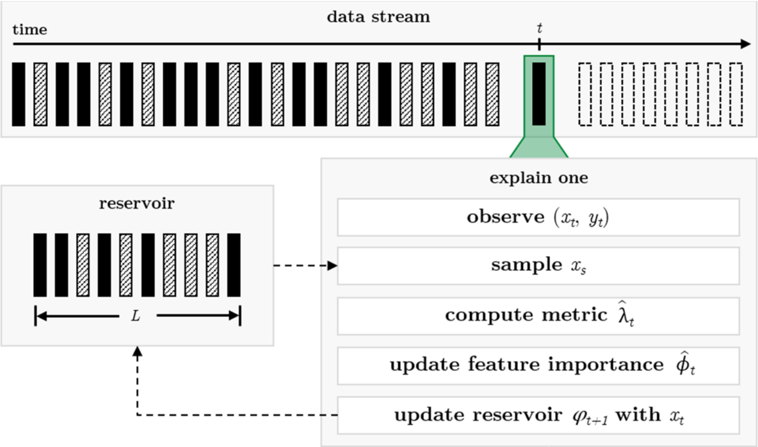

data stream
time
$t$
reservoir
$L$
explain one
observe $(x_t, y_t)$
sample $x_s$
compute metric $\hat{\lambda}_t$
update feature importance $\hat{\phi}_t$
update reservoir $\varphi_{t+1}$ with $x_t$

**Fig. 2** Illustration of the incremental explanation procedure

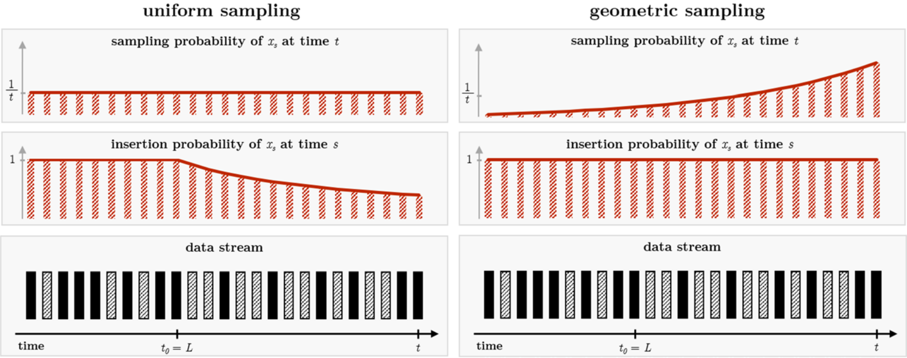

uniform sampling
geometric sampling
sampling probability of $x_s$ at time $t$
$\frac{1}{t}$
insertion probability of $x_s$ at time $s$
1
data stream
time
$t_0 = L$
$t$

**Fig. 3** Comparison of uniform (left) and geometric (right) sampling strategies. A reservoir of length $L$ summarizes the data stream (rectangles) until time $t$. The insertion probability denotes the probability that a data point is added to the reservoir at time $s$ when it is observed. The sampling probability denotes the likelihood of drawing the individual observations at time $t$

$$p_{s,r} := \mathbb{P}(\varphi_s = r) \text{ for } s = t_0, \dots, t \text{ and } r = 0, \dots, s-1, \tag{5}$$

i.e. the probability to select a previous observation from time $r$ at time $s$. Note that $t_0 > 0$ is the first time step where $\hat{\phi}_t^{(S_j)}$ can be computed, as we need previous observations for the sampling process. In the following, we assume that the sampling strategy $(\varphi_s)_{t_0 \le s \le t}$ is fixed and clear from the context, and thus omit the dependence on $\hat{\phi}_t^{(S_j)}$. We illustrate one explanation step at time $t$ in Algorithm 1 and Fig. 2. This directly corresponds to (3) with $M = 1$ and can be extended to $M > 1$ by repeatedly running the procedure in parallel and averaging the results. Next, we discuss two possible sampling strategies, which are illustrated in Fig. 3.

### 3.1 Incremental sampling strategies $\varphi$

Since random permutations cannot easily be realized in an incremental setting as they require infinite memory of previous observations and knowledge of future events, we now present two alternative types of sampling strategies. We formalize $(\varphi_s)_{t_0 \leq s \leq t}$ to choose the previous observation $r$ at time $s$ for the calculation in $\hat{\lambda}_s^{(S_j)}$. To do so, we will specify the probabilities $p_{s,r}$ in (5). An illustration of both approaches can be found in Fig. 3.

#### 3.1.1 Uniform sampling

In uniform sampling we assume that each previous observation is equally likely to be sampled at time $s$, i.e., $p_{s,r} = 1/s$ for $s = t_0, \dots, t$ and $r = 0, \dots, s-1$. It could be naively implemented by storing all previous observations and uniformly sampling at each time step. However, since memory is limited, uniform sampling may be implemented with histograms for categorical features of known and small cardinality. For others, a reservoir of fixed length $L$ can be maintained, known as reservoir sampling (Vitter, 1985). The probability of a new observation to be included in the reservoir, referred to as *insertion probability*, then decreases over time, see Fig. 3. Clearly, observations are drawn independently, but can be sampled more than once. In a data stream scenario, where changes to the underlying data distribution occur over time, the uniform sampling strategy may be inappropriate, and sampling strategies that prefer recent observations may be better suited.

#### 3.1.2 Geometric sampling

Geometric sampling arises from the idea to maintain a reservoir of size $L$, which is updated by a new observation at each time step by randomly replacing a reservoir observation with the newly observed one. Until time $t_0$ the first $L$ observations are stored in the reservoir. At each sampling step ($t \geq t_0$) an observation is uniformly chosen from the reservoir with probability $p := 1/L$. Independently, a sample from the reservoir is selected with the same probability $p := 1/L$ for replacement with the new observation. The resulting probabilities are of the geometric form $p_{s,r} = p(1-p)^{s-r-1}$ for $r \geq t_0$ and $p_{s,r} = p(1-p)^{s-t_0}$ for $r < t_0$. Clearly, the geometric sampling strategy yields increasing probabilities for more recent observations and we demonstrate in our experiments that this can be beneficial in scenarios with concept drift.

### 3.2 Theoretical results of estimation quality

The estimator $\hat{\phi}_t^{(S_j)}$ picks up the notion of the PFI estimator $\hat{\phi}^{(S_j)}$ in (3), which approximates the expectation over the random sampling strategy $(\varphi)_{t_0 \leq s \leq t}$ by averaging repeated realizations. While $\hat{\phi}_t^{(S_j)}$ only considers one realization of the sampling strategy, it is easy to extend the approach in the incremental learning scenario by computing the estimator $\hat{\phi}_t^{(S_j)}$ in multiple separate runs in parallel. While this yields an efficient estimate of PFI, it is difficult to analyze the estimator theoretically as each estimator highly depends on the realizations of the sampling strategy. We, thus, again study the expectation over the sampling strategy and introduce the expected iPFI.

**Definition 5** (*Expected iPFI*) For a data stream at time $t$ with previous observations $(x_0, y_0), \dots, (x_t, y_t)$ and a sampling strategy $\varphi := (\varphi_s)_{s=t_0,\dots,t}$ for $t_0 > 0$, we defined the *expected iPFI* as

$$\bar{\phi}_t^{(S_j)} := \mathbb{E}_\varphi[\hat{\phi}_t^{(S_j)}],$$

which corresponds to the expected PFI (model reliance) $\bar{\phi}^{(S_j)}$ in the batch setting.

To evaluate the estimation quality, we will analyze the bias $|\bar{\phi}_t^{(S_j)} - \phi^{(S_j)}(h_t)|$ and the variance of $\bar{\phi}_t^{(S_j)}$. Both can be combined by Chebyshev's inequality to obtain bounds on the approximation error of $\phi^{(S_j)}(h_t)$ for $\epsilon > |\bar{\phi}_t^{(S_j)} - \phi^{(S_j)}(h_t)|$ as

$$\mathbb{P}(|\bar{\phi}_t^{(S_j)} - \phi^{(S_j)}(h_t)| > \epsilon) = \mathcal{O}(\mathbb{V}[\bar{\phi}_t^{(S_j)}]). \tag{6}$$

As already said, all proofs are deferred to the supplementary material in Sect. A. Our theoretical results are stated and proven in a general manner, which allows one to extend our approach to other sampling strategies, other feature subsets, and even other aggregation techniques.

*Static model* Given iid observations from a data stream, we consider an incremental model that learns over time. We begin under the simplified assumption that the model does not change over time, i.e., $h_t \equiv h$ for all $t$.

**Theorem 2** (Bias for static Model) *If $h \equiv h_t$, then*

$$\phi^{(S_j)}(h) - \mathbb{E}[\bar{\phi}_t^{(S_j)}] = (1-\alpha)^{t-t_0+1}\phi^{(S_j)}(h).$$

From the above theorem it is clear that the bias of the expected iPFI $\bar{\phi}_t^{(S_j)}$ is exponentially decreasing towards zero for $t \to \infty$ and we thus continue to study the asymptotic estimator $\lim_{t\to\infty} \bar{\phi}_t^{(S_j)}$. While the bias does not depend on the sampling strategy, our next results analyzes the variance of the asymptotic estimator, which does depend on the sampling strategy.

**Theorem 3** (Variance for static Model) *If $h_t \equiv h$ and $\mathbb{V}[\|h(X_s^{(\bar{S}_j)}, X_r^{(S_j)}) - Y_s\| - \|h(X_s) - Y_s\|] < \infty$, then*

$$\text{Uniform: } \mathbb{V}\left[\lim_{t\to\infty} \bar{\phi}_t^{(S_j)}\right] = \mathcal{O}(-\alpha \log(\alpha)).$$

$$\text{Geometric: } \mathbb{V}\left[\lim_{t\to\infty} \bar{\phi}_t^{(S_j)}\right] = \mathcal{O}(\alpha) + \mathcal{O}(p).$$

The variance is therefore directly controlled by the choice of parameters $\alpha$ and $p$. As the asymptotic estimator is unbiased, it is clear that these parameters control the approximation error, as shown in (6).

*Changing model* So far, we discussed properties of $\bar{\phi}_t^{(S_j)}$ under the simplified assumption that $h_t$ does not change over time. In an incremental learning scenario, $h_t$ is updated incrementally at each time step. In cases where no concept drift affects the underlying data generating distribution, we can assume that an incremental learning algorithm gradually converges to an optimal model. We thus assume that the change of the model is controlled and show results similar to the case where $h_t$ is static. To control model change formally, we introduce $f_S^{\Delta}(x^{(\bar{S}_j)}, h_s, h_t) := \mathbb{E}_{\tilde{X}\sim\mathbb{P}_S}[\|h_t(x^{(\bar{S}_j)}, \tilde{X}) - h_s(x^{(\bar{S}_j)}, \tilde{X})\|]$. The expectation of $f_S^{\Delta}$ is denoted $\Delta_S(h_s, h_t) := \mathbb{E}_X[f_S^{\Delta}(X, h_s, h_t)]$ and $\Delta(h_s, h_t) := \Delta_{\emptyset}(h_s, h_t)$. We show that $\Delta_S$ and $\Delta$ bound the difference of FI of two models $h_t$ and $h_s$ and the bias of our estimator.

**Theorem 4** (Bias for changing Model) *If* $\Delta(h_s, h_t) \leq \delta$ *and* $\Delta_S(h_s, h_t) \leq \delta_S$ *for* $t_0 \leq s \leq t$, *then*

$$|\mathbb{E}[\bar{\phi}_t^{(S_j)}] - \phi^{(S_j)}(h_t)| \leq \delta_S + \delta + \mathcal{O}((1-\alpha)^t).$$

In the case of a changing model the estimator is therefore only unbiased if $h_t \to h$ as $t \to \infty$. For results on the variance, we control the variability of the models at different points in time. In the case of a static model, the covariances can be uniformly bounded, as they do not change over time. Instead, for a changing model, we introduce the time-dependent function

$$f_s(Z_s, Z_r) := \|h_s(X_s^{(\bar{S}_j)}, X_r^{(S_j)}) - Y_s\| - \|h_s(X_s) - Y_s\|$$

and assume existence of some $\sigma^2_{\max}$ such that

$$\mathrm{cov}(f_s(Z_s, Z_r), f_{s'}(Z_{s'}, Z_{r'})) \leq \sigma^2_{\max} \tag{7}$$

for $t_0 \leq s, s' \leq t$, $r < s$ and $r' < s'$.

**Theorem 5** (Variance for changing Model) *Given* (7) *for a sequence of models* $(h_t)_{t\geq 0}$, *the results of Theorem* 3 *apply.*

*Summary* We have shown that the approximation error of iPFI for FI is controlled by the parameters $\alpha$ and $p$. In the case of drifting data, the approximation error is additionally affected by the changes in the model, as it is then possibly biased and the covariances may change over time. As the expected PFI estimator has an approximation error of order $\mathcal{O}(1/N)$ for FI, we conclude that the above bounds on the approximation error of expected iPFI are also valid when compared with the expected PFI, if $\alpha$ is chosen according to $\alpha = \frac{2}{N+1}$. In the next section, we corroborate our theoretical findings with empirical evaluations and showcase the efficacy of iPFI in scenarios with concept drift. We also elaborate on the differences between the two sampling strategies.

## 4 Experiments

We conduct multiple experimental studies to validate our theoretical findings and present our approach on real data. We consider three benchmark datasets, which are well-established in the FI literature (Covert et al., 2020; Lundberg & Lee, 2017) called *adult* (Kohavi, 1996), *bank* (Moro et al., 2011), and *bike* (Fanaee-T & Gama, 2014), where *bike* constitutes a regression task. We further consider two binary classification real-world data streams called *elec2* (Harries, 1999) and *ozone* (de Souza et al., 2020). Moreover, we apply the multi-class *insects* (de Souza et al., 2020) data stream. Lastly, we create multiple synthetic data streams based on the *agrawal* (Agrawal et al., 1993) and *stagger* (Schlimmer & Granger, 1986) concept generators where we manually induce concept drifts. As our approach is inherently model-agnostic, we present experimental results for different model types. In the static

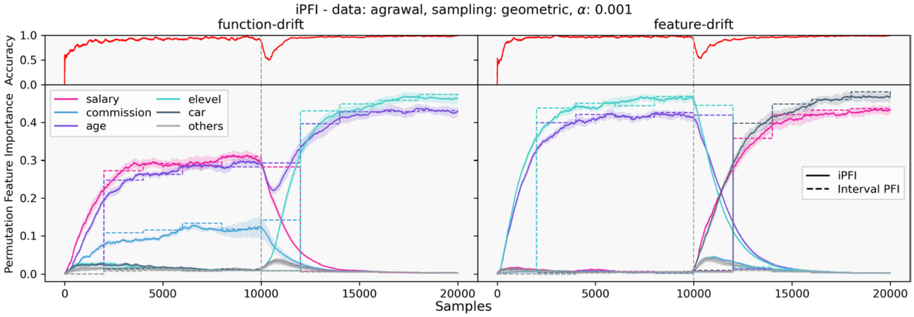


**Fig. 4** iPFI on two *agrawal* concept drift data streams for ARF classifiers. The most important features are highlighted in color. The dashed line denotes the batch calculation at set intervals (Color figure online)

batch scenario we apply Gradient Boosting Tree (GBT) (Friedman, 2001) and LightGBM (LGBM) (Ke et al., 2017) ensembles and train small 2-layer Neural Networks (NN) with layer sizes (128, 64). In the dynamic incremental learning setting, we apply Adaptive Random Forest classifiers (ARF) (Gomes et al., 2017), small scale 3-layer NNs with layer sizes (100, 100, 10) and Hoeffding Adaptive Trees (HATs) Bifet & Gavaldà (2009). The models' and data streams' implementation is based on *scikit-learn* (Pedregosa et al., 2011), *river* (Montiel et al., 2020), *PyTorch* (Paszke et al., 2017), and *OpenML* (Feurer et al., 2020). We mainly rely on default parameters, yet the supplement in Sect. C contains additional information about the datasets and details about the applied models.[3] In all our experiments, we compute the **iPFI** estimator $\hat{\phi}^{(S_j)}_{\text{iPFI}}$ as the average over ten realizations $\hat{\phi}^{(S_j)}_t$ of the incremental sampling strategies (uniform or geometric). All baseline approaches are chosen, such that they require the same amount of model evaluations as iPFI.

## 4.1 Experiment A: online PFI calculation under drift

First, we consider a dynamic modeling scenario. Here, instead of a pre-trained model, we fit different models incrementally on real data streams and compute iPFI on the fly. We incrementally train ARF, HAT and NN models. However, as our approach is inherently model-agnostic, any incremental model (implemented for example in *river*) can be explained. As a baseline, we compare our approach to the **interval PFI** for feature $j \in D$, which computes the PFI over fixed time intervals during the online learning process with ten random permutations in each interval. This can be seen as a naive implementation of iPFI with large gaps of uncertainty and a substantial time delay.

With the synthetic *agrawal* stream we induce two kinds of *real* concept drifts: First, we switch the classification function of the data generator, which we refer to as function-drift (changing the functional dependency but retaining the distribution of $X$). Second, we switch the values of two or more features with each other, which we refer to as feature-drift (changing the functional dependency by changing the distribution of $X$). Note that feature-drift can be applied to datasets, where the classification function is unknown (like *elec2*).

Figure 4 showcases how well iPFI reacts to both concept drift scenarios. Both concept drifts are induced in the middle of the data stream (after 10,000 samples). For the function-drift example (Fig. 4, left), the *agrawal* classification function was switched from Agrawal

[3] All experiments can be found at https://github.com/mmschlk/iPFI.

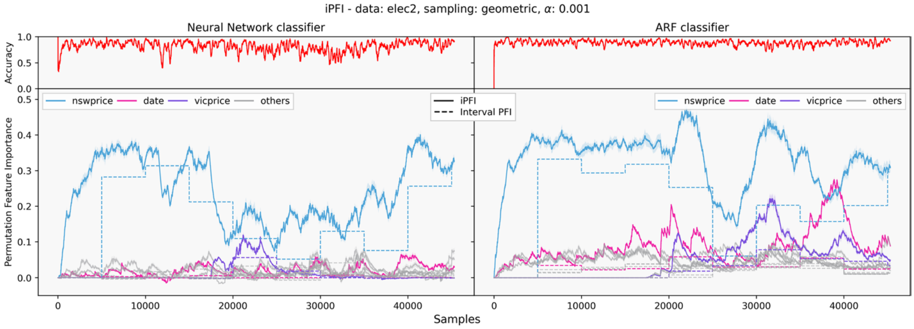


**Fig. 5** iPFI on *elec2* (without inducing a feature drift) for an incrementally fitted NN (left) and an ARF (right)

**Table 1** Summary of the additional time complexity of iPFI

| Data | *stagger* | *elec2* | *agrawal* | *adult* | *bank* | *insects* | *ozone* |
|---|---|---|---|---|---|---|---|
| Feature count | 3 | 8 | 9 | 14 | 16 | 33 | 72 |
| Explanation time | 0.734 | 1.210 | 1.411 | 1.976 | 2.386 | 5.070 | 7.717 |
| | (.017) | (.039) | (.020) | (.118) | (.048) | (.078) | (.182) |
| Inference time | 0.959 | 0.989 | 0.987 | 0.991 | 0.991 | 0.990 | 0.998 |
| | (.001) | (.002) | (.001) | (.002) | (.001) | (.021) | (.000) |

The additional *explanation time* is given relatively to the case where the models are trained without explaining. The *inference time* denotes the portion of the explanation time in which the models are queried. All values for each dataset are derived from ten independent runs. The run time of iPFI scales *linearly* with $0.104 \cdot |D|$ over the number of features ($R^2 = 0.966$)

et al. (1993)'s concept 2 to concept 3. Theoretically, only two features should be important for both concepts: For the first concept the pink *salary* and the purple *age* features are needed, and for the second concept the classification function relies on the cyan *education* and the purple *age* features. However, the ARF model also relies on the blue *commission* feature, which can be explained as *commission* directly depends on *salary* and, thus, is transitively correlated with the target variable.

In the feature-drift scenario (Fig. 4, right), the ARF model adapts to a sudden drift where both important features (*education* and *age*) are switched with two unimportant features (*car* and *salary*). In both scenarios iPFI instantly detects the shifts in importance. From both simulations, it is clear that iPFI and its anytime computation has clear advantages over interval PFI. In fact, iPFI quickly reacts to changes in the data distribution while still closely matching the "ground-truth" results of the interval-wise computation.

Next to synthetic concept drifts on *agrawal*, Fig. 5 illustrates how iPFI explanations are model-agnostic on the original *elec2* data stream. There, we incrementally train a NN and an ARF classifier on the stream without inducing an additional feature drift. For further concept drift scenarios, we refer to the supplementary material in Sect. C.

*Time complexity*

Aside from the approximation quality in the incremental setting, we also summarize the additional time complexity of iPFI in Table 1 and observe a linear relationship

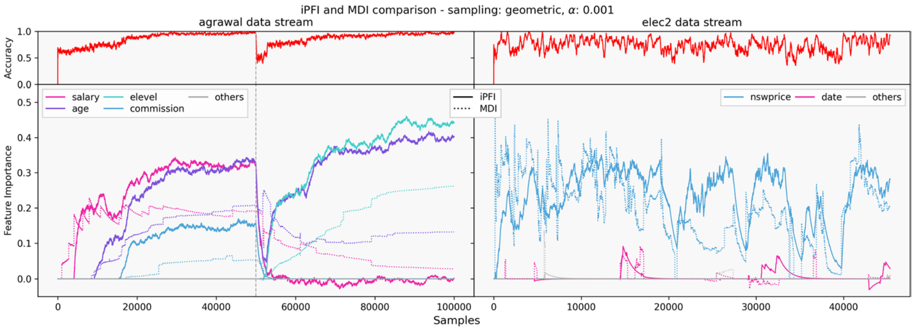


**Fig. 6** Comparison of iPFI (solid) and MDI (dotted) on an *agrawal* concept drift stream (concept 2 to 3 (Agrawal et al., 1993), left) and *elec2* (right). For each stream a single HAT classifier is trained and explained

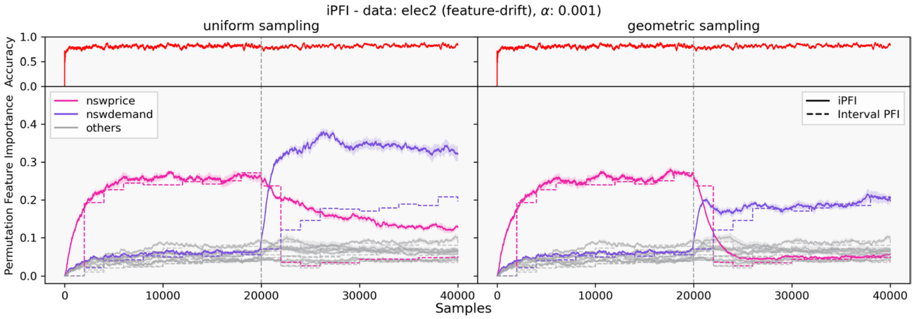


**Fig. 7** iPFI with uniform (left) and geometric sampling (right) on *elec2* with a feature-drift

$(0.104 \cdot |D|, R^2 = 0.966)$ over the feature count $|D|$. For a detailed illustration of the linear relationship we refer to Sect. C.4. We run the explanation procedure ten times for seven datasets and track the run-time with and without iPFI explanations. To isolate the variability of the run-times to the explanation procedure, we use the same ARF classification model for all seven datasets. We further decompose the explanation time into the time it takes to run the model in inference (line 3 in Algorithm 1) and the remaining storing and sampling overhead. Most of the explanation time (95% to 99%) is dedicated to the inference time of the models for which performance gains cannot be easily achieved without parallelization.

*Sanity check with tree-specific mean decrease in impurity* To further illustrate the efficacy of our approach, we also compare our model-agnostic iPFI explainer to the model-specific baseline of Mean Decrease in Impurity (MDI). Earlier works (Cassidy & Deviney, 2014; Gomes et al., 2019) leverage MDI as an importance measure in the incremental setting. Similar to Gomes et al. (2019), we manually compute the MDI on incremental summary statistics stored at each split-node of a HAT classifier. As a impurity measure, we compute the gini impurity index like in Cassidy and Deviney

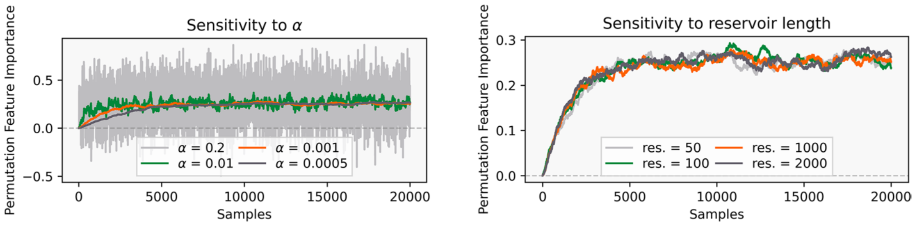


**Fig. 8** The importance of the *nswprice* feature for an ARF model training on *elec2* for different values of $\alpha$ (left) and reservoir length (right)

**Table 2** Median error of iPFI compared to batch PFI (IQR between $Q_1$ and $Q_3$ in braces)

| Data (N) | **Model** (perf.) | **Error** | |
|---|---|---|---|
| | | Uniform | Geometric |
| *agrawal* (20k) | LGBM (99%) | 0.011 (.006) | 0.010 (.006) |
| *elec2* (≈45k) | LGBM (88%) | 0.038 (.012) | 0.037 (.011) |
| *adult* (≈45k) | GBT (86%) | 0.126 (.040) | 0.114 (.025) |
| *bank* (≈45k) | NN (91%) | 0.126 (.024) | 0.132 (.013) |
| *bike* (≈17k) | LGBM (26.6) | 0.022 (.005) | 0.019 (.008) |

Model performance is measured in accuracy and MAE (*bike*)

(2014). Figure 6 shows the comparison of iPFI and MDI for an *agrawal* concept drift data stream and *elec2*. Aside from the differing scales, both measures detect the same importance rankings and react to concept drift. However, as MDI can only be computed for tree-based models such as HATs and ARFs, its applicability is strictly limited compared to the model-agnostic approach of calculating iPFI, which can be applied to any model class and loss function.

### 4.2 Experiment B: Geometric vs. uniform sampling

Second, we focus on the question, which sampling strategy to prefer in which learning environments. We conclude that geometric sampling should be applied under feature-drift scenarios, as the choice of sampling strategy substantially impacts iPFI's performance in concept drift scenarios where feature distributions change over time. If a dynamic model adapts to changing feature distributions, and the PFI is estimated with samples from the outdated distribution, the resulting replacement samples are outside the current data manifold. Estimating PFI by using this data can result in skewed estimates, as illustrated in Fig. 7. There, we induce a feature-drift by switching the values of the most important feature for an ARF model on *elec2* with a random feature. The uniform sampling strategy (Fig. 7, left) is incapable of matching the "ground-truth" interval PFI estimation like the geometric sampling strategy (Fig. 7, right). Hence, in dynamic learning environments like data stream analytics or continual learning, we recommend applying a sampling strategy that focuses on more recent samples, such as geometric distributions. For applications without drift in the feature-space like progressive data science, uniform sampling strategies, which evenly distribute the probability of a data point being sampled across the data stream, may still be preferred.

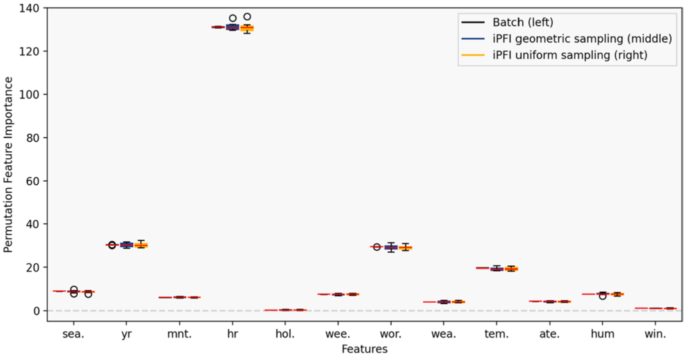


**Fig. 9** Boxplot of PFI estimates per feature of the *bike* regression dataset for batch PFI (left), geometric sampling iPFI (middle), and uniform sampling iPFI (right) on a pre-trained LGBM regressor

*Parameter considerations* We, further, conduct an analysis of the two most important hyperparameters on the *elec2* data stream. The results are shown in Fig. 8. Therein, we show that the smoothing parameter $\alpha$ substantially effects iPFI's FI estimates. Like any smoothing mechanism, this parameter controls the deviation of iPFI's estimates. This parameter should be set individually for the task at hand. In our experiment, values between $\alpha = 0.001$ (conservative) and $\alpha = 0.01$ (reactive) appeared to be reasonable. The size of the the reservoir does not substantially effect the estimation quality for values between 50 and 2 000.

### 4.3 Experiment C: Approximation of batch PFI

We further consider the static model setting where models are pre-trained before they are explained on the whole dataset (no incremental learning). This experiment demonstrates that iPFI correctly approximates batch PFI estimation. We compare iPFI with the classical **batch PFI** $\hat{\phi}^{(S_j)}_{\text{batch}}$ for feature $j \in D$, which is computed using the whole static dataset over ten random permutations. We normalize $\hat{\phi}^{(S_j)}_{\text{iPFI}}$ and $\hat{\phi}^{(S_j)}_{\text{batch}}$ between 0 and 1, and compute the sum over the feature-wise absolute approximation errors $\sum_{j \in D} |\hat{\phi}^{(S_j)}_{\text{iPFI}} - \hat{\phi}^{(S_j)}_{\text{batch}}|$. Table 2 shows the median and interquartile range (IQR) (difference between the first and third quartile) of the error based on ten random orderings of each dataset. Figure 9 illustrates the approximation quality of iPFI with geometric and uniform sampling per feature for the *bike* regression dataset. Further results can be found in the supplement material in Sect. C. In the static modeling case, there is no clear difference between geometric and uniform sampling. However, in the dynamic modeling context under drift, the sampling strategy has a substantial effect on the iPFI estimates.

## 5 Conclusion and future work

In this work, we considered global FI as a statistic measure of change in the model's risk when features are marginalized. We discussed PFI as an approach to estimate feature importance and proved that only appropriately scaled permutation tests are unbiased estimators of global FI (Theorem 1). In this case, the expectation over the sampling strategy (*expected PFI*) then corresponds to the model reliance U-Statistic (Fisher et al., 2019).

Based on this notion, we presented iPFI, which is a model-agnostic algorithm to incrementally estimate global FI with PFI by averaging importance scores for individual observations over repeated realizations of a sampling strategy. We introduced two incremental sampling strategies and established theoretical results for the expectation over the sampling strategy (*expected iPFI*) to control the approximation error using iPFI's parameters. On various benchmark datasets, we demonstrated the efficacy of our algorithms by comparing them with the batch PFI baseline method in a static progressive setting as well as with interval-based PFI in a dynamic incremental learning scenario with different types of concept drift and parameter choices.

Applying XAI methods incrementally to data stream analytics offers unique insights into models that change over time. In this work, we rely on PFI as an established and inexpensive FI measure. Other computationally more expensive approaches (such as SAGE) address some limitations of PFI. As our theoretical results can be applied to arbitrary feature subsets, analyzing these methods in the dynamic environment offers interesting research opportunities. In contrast to this work's technical focus, analyzing the dynamic XAI scenario through a human-focused lens with human-grounded experiments is paramount (Doshi-Velez & Kim, 2017).

## Organisation of the appendix

The supplement material is organized as follows. Section A contains all proofs of the theoretical analysis conducted in the main body of the work. Section B covers the approximation error of expected PFI. Further experimental results and detailed descriptions of the datasets and models used for the empirical analysis is discussed in Sect. C. Lastly, Sect. D shows how PFI may be computed analytically for a pre-defined classification function illustrated with the *agrawal* concepts.

## A Proofs

In the following, we provide the proofs of all theorems. We further present more general results that are stated as propositions.

**Theorem 6** *The expected PFI* (*model reliance*) *can be rewritten as a normalized expectation over uniformly random permutations*, *i.e.*

$$\bar{\phi}^{(S_j)} = \frac{N}{N-1}\mathbb{E}_{\varphi\sim\text{unif}(\mathfrak{S}_N)}\left[\hat{\phi}_{\varphi}^{(S_j)}\right] \approx \hat{\phi}^{(S_j)} \tag{8}$$

*i.e. expected PFI is canonically estimated by the PFI estimator and in particular* $\bar{\phi}^{(S_j)} = \mathbb{E}_\varphi[\hat{\phi}^{(S_j)}]$.

***Proof*** We write $f(z_n, z_m) := \|h(x_n^{(\bar{S}_j)}, x_m^{(S_j)}) - y_n\| - \|h(x_n) - y_n\|$ and compute the expectation over randomly sampled permutations $\varphi \in \mathfrak{S}_N$. Each permutation has probability $\frac{1}{N!}$, which yields

$$\begin{aligned}
\mathbb{E}_\varphi[\hat{\phi}_\varphi^{(S_j)}] &= \frac{1}{N!} \sum_{\varphi \in \mathfrak{S}_N} \hat{\phi}_\varphi^{(S_j)} \\
&= \frac{1}{N}\frac{1}{N!} \sum_{n=1}^{N} \sum_{\varphi \in \mathfrak{S}_N} f(z_n, z_{\varphi(n)}) \\
&= \frac{1}{N}\frac{1}{N!} \sum_{n=1}^{N} \sum_{m=1}^{N} (N-1)! f(z_n, z_m) \\
&= \frac{1}{N}\frac{1}{N} \sum_{n=1}^{N} \sum_{m \neq n} f(z_n, z_m) \\
&= \frac{1}{N}\frac{1}{N} \sum_{n=1}^{N} \sum_{m \neq n} \|h(x_n, x_m) - y_n\| \\
&- \frac{N-1}{N^2} \sum_{n=1}^{N} \|h(x_n) - y_n\|,
\end{aligned}$$

where we used in the third line that there are $(N-1)!$ permutations with $\varphi(n) = m$. We thus conclude,

$$\frac{N}{N-1}\mathbb{E}_\varphi[\hat{\phi}_\varphi^{(S_j)}] = \hat{e}_{\text{switch}} - \hat{e}_{\text{orig}} = \bar{\phi}^{(S_j)}.$$

□

**Theorem 7** (Bias for static Model) *If* $h \equiv h_t$, *then*

$$\phi^{(S_j)}(h) - \mathbb{E}[\bar{\phi}_t^{(S_j)}] = (1-\alpha)^{t-t_0+1}\phi^{(S_j)}(h).$$

***Proof*** We consider the more general estimator $\tilde{\phi}_t^{(S)} := \mathbb{E}_\varphi[\sum_{s=t_0}^t w_s \hat{\lambda}_t^{(S)}(x_t, x_{\varphi_t}, y_t)]$ and prove a more general result that can be used for arbitrary sampling and aggregation techniques.

**Proposition 8** *If* $h \equiv h_t$, *then*

$$\phi^{(S)}(h) - \mathbb{E}[\tilde{\phi}_t^{(S)}] = (1-\mu_w)\phi^{(S)}(h)$$

*with* $\mu_w := \sum_{s=t_0}^t w_s$.

***Proof*** As each $\hat{\lambda}_s^{(S)}$ is an unbiased estimator of $\phi^{(S)}(h_s)$, we have $\mathbb{E}[\tilde{\phi}_t^{(S)}] = \sum_{s=t_0}^t w_s \phi^{(S)}(h) = \mu_w \phi^{(S)}(h)$, where we used $(\varphi)_{t_0 \le s \le t} \perp (X, Y)$. □

The result then follows directly, as $\bar{\phi}^{(S)} = \tilde{\phi}^{(S)}$ for $w_s := \alpha(1-\alpha)^{t-s}$, $\mu_w = 1-(1-\alpha)^{t-t_0+1}$ and $S := S_j$. □

**Theorem 9** (Variance for static Model) *If* $h_t \equiv h$ *and* $\mathbb{V}[\|h(X_s^{(\bar{S}_j)}, X_r^{(S_j)}) - Y_s\| - \|h(X_s) - Y_s\|] < \infty$, *then*

$$\text{Uniform: } \mathbb{V}\left[\lim_{t\to\infty} \bar{\phi}_t^{(S_j)}\right] = \mathcal{O}(-\alpha\log(\alpha)).$$
$$\text{Geometric: } \mathbb{V}\left[\lim_{t\to\infty} \bar{\phi}_t^{(S_j)}\right] = \mathcal{O}(\alpha) + \mathcal{O}(p).$$

***Proof*** We again consider the more general estimator $\tilde{\phi}_t^{(S)} := \mathbb{E}_\varphi[\sum_{s=t_0}^t w_s \hat{\lambda}_t^{(S)}(x_t, x_{\varphi_t}, y_t)]$ and prove a result, that can be used for arbitrary sampling and aggregation techniques.

□

**Proposition 10** *For* $\varphi$ *from* (5) *with* $\varphi_s \perp \varphi_r$ *for* $r < s$ *and* $p_{s,r} \le p_{s',r}$ *for* $s > s'$, *i.e., the probability to sample a previous observation r is non-increasing over time, it holds*

$$\mathbb{V}\left[\tilde{\phi}_t^{(S)}\right] \le 4\sigma_w^2\sigma_2^2 + 2\sigma_2^2 \sum_{s=t_0}^{t}\sum_{s'=t_0}^{s-1} w_s w_{s'} \underbrace{\sum_{r=0}^{s'-1} p_{s',r}^2}_{=:\mathcal{I}_\varphi(s)},$$

*provided that* $\sigma_2^2 := \mathbb{V}[f(Z_s, Z_r)] < \infty$ *and with* $\sigma_w^2 := \sum_{s=0}^t w_s^2$.

***Proof*** We denote $f(Z_s, Z_r) := \|h(X_s^{(\bar{S}_j)}, X_r^{(S)}) - Y_s\| - \|h(X_s) - Y_s\|$. Using $p_{s,r} := \mathbb{P}(\varphi_s = r)$ and properties of variance, we can write

$$\mathbb{V}[\tilde{\phi}_t] = \mathbb{V}[\sum_{s=t-N+1}^{t} w_s \sum_{r=0}^{s-1} p_{r,s} f(Z_s, Z_r)]$$
$$= \sum_{s,s'=t_0}^{t} w_s w_{s'} \sum_{r=0}^{s-1}\sum_{r'=0}^{s'-1} p_{s,r}p_{s',r'} \operatorname{cov}((s,r),(s',r')),$$

where $\operatorname{cov}((s,r),(s',r')) := \operatorname{cov}(f(Z_s, Z_r), f(Z_{s'}, Z_{r'}))$ denotes the covariance of the two random variables. The above sum ranges over all possible combinations of pairs $(s, r)$, where $s = t_0 \dots, t$ and $r = 0, \dots, s-1$. As $r < s$ and $r' < s'$, it holds $|\{s, s', r, r'\}| \ge 2$. When $|\{s, s', r, r'\}| = 2$ then $s = s'$ and $r = r'$ and the covariance reduces to the variance. When none of the indices match, i.e., $|\{s, s', r, r'\}| = 4$, then the covariance is zero, due to the independence assumption. When exactly one index matches, then there are three possible cases:

- Case 1: $s = s', r \neq r'$,
- Case 2: $s \neq s', r \neq r$ with $r' = s$ or $s' = r$
- Case 3: $s \neq s', r = r'$.

Case 2 yields the same covariances due to the iid assumption and the symmetric of the covariance. For case 1, with $\mathbb{E}_{(Z_s,Z_r)}[f(Z_s, Z_r)] = \mathbb{E}_{Z_s}\mathbb{E}_{Z_r}[f(Z_s, Z_r)] = \phi^{(S)}(h)$, we denote $\tilde{f}(Z_s, Z_r) := f(Z_s, Z_r) - \phi^{(S)}(h)$ to compute the covariance as

$$\begin{aligned}\operatorname{cov}((s,r),(s',r')) &= \mathbb{E}[\tilde{f}(Z_s,Z_r)\tilde{f}(Z_s,Z_{r'})]\\ &= \mathbb{E}_{Z_s}[\mathbb{E}_{Z_r}[\tilde{f}(Z_s,Z_r)]\mathbb{E}_{Z_{r'}}[\tilde{f}(Z_s,Z_{r'})]]\\ &= \mathbb{E}_{Z_s}[\mathbb{E}_{Z_r}[\tilde{f}(Z_s,Z_r)]^2]\\ &= \mathbb{V}_{Z_s}[\mathbb{E}_{Z_r}[f(Z_s,Z_r)]],\end{aligned}$$

where we have used $\mathbb{E}_{Z_s}[\mathbb{E}_{Z_r}[\tilde{f}(Z_s,Z_r)]] = \phi^{(S)}(h)$ as well as the iid assumption multiple times, in particular when $\mathbb{E}_{Z_r}[f(Z_s,Z_r)] = \mathbb{E}_{Z_{r'}}[f(Z_s,Z_{r'})]$. The same arguments apply for the second argument for case 3, as

$$\operatorname{cov}((s,r),(s',r')) = \mathbb{V}_{Z_r}[\mathbb{E}_{Z_s}[f(Z_s,Z_r)]].$$

We thus summarize

$$\operatorname{cov}((s,r),(s',r')) = \begin{cases}\mathbb{V}[f(Z_s,Z_r)], \text{ if } s=s', r=r'\\ \mathbb{V}_{Z_s}[\mathbb{E}_{Z_r}[f(Z_s,Z_r)]], \text{ if case 1}\\ \operatorname{cov}((s,r),(s',r')), \text{ if case 2}\\ \mathbb{V}_{Z_r}[\mathbb{E}_{Z_s}[f(Z_s,Z_r)]], \text{ if case 3}\\ 0, \text{ if } |\{s,s',r,r'\}| = 4.\end{cases}$$

By the Cauchy–Schwarz inequality all covariances are bounded by $\sigma_2^2 := \mathbb{V}[f(Z_s,Z_r)]$. With $I := \{t_0,\dots,t\}$ and $I_s := \{0,\dots,s-1\}$ and $Q_2 := \{(s,r) : s = s' \in I, r = r' \in I_s\}$ $Q_3 := \{(s,s',r,r') : s,s' \in I, r \in I_s, r' \in I_{r'}, |\{s,s',r,r'\}| = 3\}$. We thus obtain

$$\begin{aligned}\mathbb{V}[\tilde{\phi}_t^{(S)}] = \sigma_2^2 &\sum_{(s,r)\in Q_2} w_s^2 p_{s,r}^2\\ &+ \sum_{(s,s',r,r')\in Q_3} w_s w_{s'} p_{s,r} p_{s',r'} \operatorname{cov}((s,r),(s',r')).\end{aligned}$$

For the first sum, we have

$$\sum_{(s,r)\in Q_2} w_s^2 p_{s,r}^2 \leq \sum_{(s,r)\in Q_2} w_s^2 p_{s,r} = \sum_{s=t_0}^{t} w_s^2 = \sigma_w^2.$$

For the second sum, $Q_3$ decomposes into the three cases. For case 1,

$$\begin{aligned}\sum_{\substack{(s,s',r,r')\in Q_3\\ s=s', r\neq r'}} w_s w_{s'} p_{s,r} p_{s,r'} &= \sum_{s=t_0}^{t} w_s w_{s'} \sum_{\substack{(r,r')\in I_s^2\\ r\neq r'}} p_{s,r} p_{s,r'}\\ &\leq \sum_{s=t_0}^{t} w_s^2 (\sum_{r=0}^{s-1} p_{s,r})^2 = \sigma_w^2.\end{aligned}$$

For case 2 w.l.o.g assume $r = s'$, which implies $s > s'$ and thus $w_s \geq w_{s'}$, then

$$\sum_{\substack{(s,s',r,r')\in Q_3\\ s\neq s', r\neq r', s'=r}} w_s w_{s'} p_{s,s'} p_{s',r'} = \sum_{s=t_0}^{t} w_s \sum_{s'=t_0}^{s-1} w_{s'} p_{s,s'}$$

$$\leq \sum_{s=t_0}^{t} w_s^2 = \sigma_w^2.$$

For case 3, we have

$$\begin{aligned}\sum_{\substack{(s,s',r,r')\in Q_3\\ s\neq s', r=r'}} w_s w_{s'} p_{s,r} p_{s',r} &= \sum_{\substack{(s,s')\in I^2\\ s\neq s'}} w_s w_{s'} \sum_{r=0}^{\min(s,s')-1} p_{s,r} p_{s',r}\\ &= 2 \sum_{\substack{(s,s')\in I^2\\ s> s'}} w_s w_{s'} \sum_{r=0}^{s'-1} p_{s,r} p_{s',r}\\ &\leq 2 \sum_{\substack{(s,s')\in I^2\\ s> s'}} w_s w_{s'} \sum_{r=0}^{s'-1} p_{s',r}^2.\end{aligned}$$

In summary, we conclude

$$\mathbb{V}\left[\tilde{\phi}_t^{(S)}\right] \leq 4\sigma_w^2\sigma_2^2 + 2\sigma_2^2 \sum_{s=t_0}^{t}\sum_{s'=t_0}^{s-1} w_s w_{s'} \sum_{r=0}^{s'-1} p_{s',r}^2.$$

□

The last sum depends on both the choices of weights $w_s$ and the *collision probability* $\mathcal{I}_\varphi(s) = \sum_{r=0}^{s-1} p_{s,r}^2 = P(Q_1 = Q_2)$ for $Q_1, Q_2 \overset{iid}{\sim} \mathbb{P}_{\varphi_s}$, which is related to the Rényi entropy (Rényi, 1961). The variance increases with the collision probabilities of the sampling strategy, in particular $\mathcal{I}_{\text{unif}}(s) = \frac{1}{s}$ and $\mathcal{I}_{\text{geom}}(s) = \frac{p}{2-p}(1 + (1-p)^{2(s-t_0)+1})$ for uniform and geometric sampling, respectively.

**Lemma 1** *For geometric sampling and* $p \in (0,1)$ *it holds*

$$\mathcal{I}_{\text{geom}}(s) = \sum_{r=0}^{s-1} p_{s,r}^2 = \frac{p}{2-p}(1 + (1-p)^{2(s-t_0)+1}).$$

***Proof*** The probabilities for geometric sampling are

$$p_{s,r} = \begin{cases} p\cdot(1-p)^{s-r-1}, r > t_0 = \frac{1}{p}\\ p\cdot(1-p)^{s-t_0}, r \leq t_0 = \frac{1}{p}.\end{cases}$$

Then

$$
\begin{aligned}
\mathcal{I}_{\text{geom}}(s) &= \sum_{r=0}^{s-1} p_{s,r}^2 \\
&= \sum_{r=0}^{t_0-1} p^2 \cdot (1-p_r)^{2(s-t_0)} + \sum_{r=t_0}^{s-1} p^2 (1-p)^{2(s-r-1)} \\
&= t_0 \cdot p^2 \cdot (1-p)^{2(s-t_0)} + \sum_{r=t_0}^{s-1} p^2 (1-p)^{2(s-r-1)} \\
&= p \cdot (1-p)^{2(s-t_0)} + p^2 \sum_{r=0}^{s-t_0-1} (1-p)^{2r} \\
&= p \cdot (1-p)^{2(s-t_0)} + p^2 \frac{1-(1-p)^{2(s-t_0)}}{1-(1-p)^2} \\
&= p \cdot (1-p)^{2(s-t_0)} + \frac{p}{2-p}(1-(1-p)^{2(s-t_0)}) \\
&= \frac{p}{2-p}(1+(1-p)^{2(s-t_0)+1}).
\end{aligned}
$$

□

We now apply Proposition 10 to our particular estimator $\bar{\phi}^{(S)} = \tilde{\phi}^{(S)}$ with $w_s := \alpha(1-\alpha)^{t-s}$ and take the limit for $t \rightarrow \infty$. Note that both uniform and geometric sampling fulfill the condition of the theorem. Furthermore, we have $\sigma_w^2 = \alpha^2 \sum_{s=0}^{t-t_0} (1-\alpha)^s \nearrow \frac{\alpha}{2-\alpha}$.

*Uniform sampling* For uniform sampling, we have

$$
\begin{aligned}
\mathbb{V}[\bar{\phi}_t^{(S)}] &\leq \frac{\alpha}{2-\alpha} 4\sigma_2^2 + 2\sigma_2^2 \sum_{s=t_0}^{t} \sum_{s'=t_0}^{s-1} \alpha^2 \frac{(1-\alpha)^{t-s+t-s'}}{s'} \\
&\leq \frac{\alpha}{2-\alpha} 4\sigma_2^2 + 2\sigma_2^2 \alpha^2 \sum_{s=0}^{t-t_0} (1-\alpha)^s \sum_{s'=0}^{t-t_0} \frac{(1-\alpha)^{s'}}{t-s'}
\end{aligned}
$$

For the first sum, we have $\alpha \sum_{s=0}^{t-t_0} (1-\alpha)^s \nearrow 1$ for $t \rightarrow \infty$. For the second sum

$$
\begin{aligned}
\alpha \sum_{s'=0}^{t-t_0} \frac{(1-\alpha)^{s'}}{t-s'} &\leq \alpha\Big( \sum_{\substack{s'=0 \\ s' \geq t/2}}^{t-t_0} (1-\alpha)^{s'} + 1 + \sum_{\substack{s'=1 \\ s' < t/2}}^{t-t_0} \frac{(1-\alpha)^{s'}}{s'} \Big) \\
&\leq (1-\alpha)^{t/2} - (1-\alpha)^{t-t_0+1} + \alpha - \alpha \log(\alpha) \\
&\overset{t\rightarrow\infty}{\longrightarrow} \alpha - \alpha \log(\alpha).
\end{aligned}
$$

Hence,

$$
\mathbb{V}[\lim_{t\rightarrow\infty} \bar{\phi}_t^{(S)}] = \mathcal{O}(-\alpha \log(\alpha)).
$$

*Geometric sampling* For geometric sampling, we have

$$\mathbb{V}[\bar{\phi}_t^{(S)}] \leq \underbrace{\frac{\alpha}{2-\alpha}4\sigma_2^2}_{=\mathcal{O}(\alpha)} + 2\sigma_2^2 \underbrace{\alpha^2 \sum_{s=t_0}^{t} \sum_{s'=t_0}^{s-1} (1-\alpha)^{t-s+t-s'}]}_{=:q(\alpha)} \mathcal{I}_{\text{geom}}(s).$$

For the second term it is enough to show that $0 < \lim_{t\to\infty} q(\alpha) < \infty$ to prove the result, as $\mathcal{I}_{\text{geom}}(s) = \mathcal{O}(p)$. By using the properties of geometric progression, we obtain

$$\begin{aligned}
q(\alpha) &= \alpha \sum_{s=t_0}^{t} (1-\alpha)^{t-s} \alpha \sum_{s'=t-s}^{t-t_0} (1-\alpha)^{s'} \\
&= \alpha \sum_{s=t_0}^{t} (1-\alpha)^{t-s} ((1-\alpha)^{t-s} - (1-\alpha)^{t-t_0+1}) \\
&= \alpha \sum_{s=0}^{t-t_0} (1-\alpha)^{s} ((1-\alpha)^{s} - (1-\alpha)^{t-t_0+1}) \\
&= \underbrace{\alpha \sum_{s=0}^{t-t_0} (1-\alpha)^{2s}}_{\nearrow \frac{1}{2-\alpha}} - (1-\alpha)^{t-t_0+1} \underbrace{\alpha \sum_{s=0}^{t-t_0} (1-\alpha)^{s}}_{\nearrow 1} \\
&\overset{t\to\infty}{\longrightarrow} \frac{1}{2-\alpha}.
\end{aligned}$$

Hence,

$$\mathbb{V}[\lim_{t\to\infty} \bar{\phi}_t^{(S)}] \leq \mathcal{O}(\alpha) + 2\sigma_2^2 \frac{2}{2-\alpha}\frac{p}{2-p} = \mathcal{O}(\alpha) + \mathcal{O}(p).$$

□

**Theorem 11** (Bias for changing Model) *If $\Delta(h_s, h_t) \leq \delta$ and $\Delta_S(h_s, h_t) \leq \delta_S$ for $t_0 \leq s \leq t$, then*

$$|\mathbb{E}[\bar{\phi}_t^{(S_j)}] - \phi^{(S_j)}(h_t)| \leq \delta_S + \delta + \mathcal{O}((1-\alpha)^t).$$

***Proof*** We again consider the more general estimator $\tilde{\phi}_t^{(S)} := \mathbb{E}_\varphi[\sum_{s=t_0}^{t} w_s \hat{\lambda}_t^{(S)}(x_t, x_{\varphi_t}, y_t)]$ and prove a more general result.

**Proposition 12** *If $\Delta(h_s, h_t) \leq \delta$ and $\Delta_S(h_s, h_t) \leq \delta_S$ for $t_0 \leq s \leq t$, then $|\mathbb{E}[\hat{\phi}_t^{(S)}] - \phi^{(S)}(h_t)| \leq \mu_w(\delta_S + \delta) + |(1-\mu_w)\phi^{(S)}(h_t)|$.*

***Proof*** For the proof, we first show that for two models $h_s, h_t$ and a subset $S \subset D$, it holds that $|\phi^{(S)}(h_t) - \phi^{(S)}(h_s)| \leq \Delta_S(h_s, h_t) + \Delta(h_s, h_t)$. This follows directly from the reverse triangle inequality for $f_S^\Delta(x^{(\bar{S})}, h_s, h_t) \geq \mathbb{E}_{\tilde{X}}[\|h_t(x^{(\bar{S})}, \tilde{X}) - y\| - \|y - h_s(x^{(\bar{S})}, \tilde{X})\|]$. The result then follows directly by definition, the observation that $\hat{\lambda}_s^{(S)}$ is an unbiased estimate of $\phi^{(S)}(h_s)$, as

$$\begin{aligned}|\mathbb{E}[\bar{\phi}_t^{(S)}] - \phi^{(S)}(h_t)| &= |\left(\sum_{s=t_0}^{t} w_s \phi^{(S)}(h_s)\right) - \phi^{(S)}(h_t)| \\ &\leq \sum_{s=t_0}^{t} w_s \underbrace{|\phi^{(S)}(h_s) - \phi^{(S)}(h_t)|}_{\leq \delta + \delta_S} \\ &\quad + |\left(\sum_{s=t_0}^{t} w_s - 1\right)\phi^{(S)}(h_t)| \\ &\leq \mu_w(\delta + \delta_S) + \underbrace{|(1 - \mu_w)\phi^{(S)}(h_t)|}_{\text{bias for static model}} .\end{aligned}$$

□

With $\mu_w = 1 - (1 - \alpha)^{t-t_0+1}$ our special case follows immediately. □

**Theorem 13** (Variance for changing Model) *If*

$$\operatorname{cov}(f_s(Z_s, Z_r), f_{s'}(Z_{s'}, Z_{r'})) \leq \sigma^2_{\max} \tag{9}$$

*for* $t_0 \leq s, s' \leq t$, $r < s$ *and* $r' < s'$, *then for a sequence of models* $(h_t)_{t \geq 0}$ *the results of Theorem* 3 *apply.*

***Proof*** In all proofs a changing model $h_t$ adds a time dependency on the function $f_s(Z_s, Z_r) := \|h_s(X_s^{(\bar{S})}, X_r^{(S)}) - Y_s\| - \|h_s(X_s) - Y_s\|$. Instead of bounding the covariances by $\sigma_2^2$, we now bound the covariances of the time-dependent functions by $\sigma^2_{\max}$. This only directly affects Proposition 10, as

$$\begin{aligned}\mathbb{V}[\bar{\phi}_t^{(S)}] &= \mathbb{V}\left[\sum_{s=t-N+1}^{t} w_s \sum_{r=0}^{s-1} p_{r,s} f_s(Z_s, Z_r)\right] \\ &= \sum_{s,s'=t_0}^{t} w_s w_{s'} \sum_{r=0}^{s-1} \sum_{r'=0}^{s'-1} p_{s,r} p_{s',r'} \operatorname{cov}((s,r),(s',r')) \\ &\leq \sigma^2_{\max} \sum_{s,s'=t_0}^{t} w_s w_{s'} \sum_{r=0}^{s-1} \sum_{r'=0}^{s'-1} p_{s,r} p_{s',r'} .\end{aligned}$$

All remaining arguments and proofs are still valid for a changing model due to the iid assumption. □

## B Approximation error for expected PFI

With $f(Z_n, Z_m) := \|h(X_n^{(\bar{S}_j)}, X_m^{(S_j)}) - Y_n\| - \|h(X_n) - Y_n\|$ and symmetric U-statistic kernel $f_0(Z_n, Z_m) := \frac{f(Z_n,Z_m)+f(Z_m,Z_n)}{2}$, we can write

$$\bar{\phi}^{(S_j)} = \binom{N}{2}^{-1} \sum_{1\leq n<m\leq N} f_0(Z_n, Z_m),$$

which is the basic form of a U-statistic and therefore the variance can be computed as

$$\mathbb{V}\left[\bar{\phi}^{(S_j)}\right] = \binom{N}{2}^{-1} \sum_{c=1}^{2} \binom{2}{c}\binom{N-2}{2-c}\sigma_c^2 = \mathcal{O}(1/N),$$

where $\sigma_1^2 := \mathbb{V}_{Z_n}[\mathbb{E}_{Z_m}[f_0(Z_n, Z_m)]]$ and $\sigma_2^2 := \mathbb{V}[f_0(Z_n, Z_m)]$ are assumed to be finite (Hoeffding, 1948). For $\epsilon > 0$, we then obtain by Chebyshev's inequality $\mathbb{P}(|\bar{\phi}^{(S_j)} - \phi^{(S_j)}(h)| > \epsilon) = \mathcal{O}(1/N)$, as $\bar{\phi}^{(S_j)}$ is unbiased.

## C Experiments

In the following, we give more comprehensive details about the datasets and models used in our experiments.

### C.1 Dataset description

*Adult* (Kohavi, 1996) Binary classification dataset that classifies 48,842 individuals based on 14 features into yearly salaries above and below 50k. There are six numerical features and eight nominal features.

*Bank* (Moro et al., 2011) Binary classification dataset that classifies 45,211 marketing phone calls based on 17 features to decide whether they decided to subscribe a term deposit. There are seven numerical features and ten nominal features.

*Bike* (Fanaee-T & Gama, 2014) Regression dataset that collects the number of bikes in different bike stations of Toulouse over 187,470 time stamps. There are six numerical features and two nominal features.

*elec2* (Harries, 1999) Binary classification dataset that classifies, if the electricity price will go up or down. The data was collected for 45,312 time stamp from the Australian New South Wales Electricity Market and is based on eight features, six numerical and two nominal.

*agrawal* (Agrawal et al., 1993) Synthetic data stream generator to create binary classification problems to decide whether an indivdual will be granted a loan based on nine features, six numerical and three nominal. There are ten different decision functions available.

*stagger* (Schlimmer & Granger, 1986) The *stagger* concepts makes a simple toy classification data stream. The syntethtical data stream generator consists of three independent categorical features that describe the *shape*, *size*, and *color* of an artificial object. Different classification functions can be derived from these sharp distinctions.

*insects* (de Souza et al., 2020) The *insects* concept drift data streams capture flight information about different kinds of mosquito in various experimental settings. In total, 11 different variants of this stream (i.e. experimental settings) are available. The streams were created in a synthetic experiment with real mosquitoes and sensors. The data stream captures flight information about different mosquito kinds in various experimental settings. The dataset contains 33 numerical features. The variant called “abrupt balanced” used here contains 52, 848 samples.

*ozone* (de Souza et al., 2020) The *ozone* dataset contains air measurements values in the years of 1998 to 2004. The learning task is a binary classification problem of determining the ozone level (“ozone” day or “normal” day). In total the dataset contains 72 numerical features for 2 534 days.

### C.2 Model description

All models are implemented with the default parameters from *scikit-learn* (Pedregosa et al., 2011) and *River* (Montiel et al., 2020) unless otherwise stated.

*ARF* The Adaptive Random Forest Classifier (ARF) uses an ensemble of 50 trees with binary splits, ADWIN drift detection and information gain split criterion. We used the default implementation *AdaptiveRandomForestClassifier* from River with n_models=50 and binary_split=True.

*NN* The Neural Network classifier (NN) was implemented with two hidden layers of size $128 \times 64$, ReLu activation function and optimized with stochastic gradient descent (ADAM). We used the default implementation *MLPClassifier* from scikit-learn.

*GBT* The Gradient Boosting Tree (GBT) uses 200 estimators and additively builds a decision tree ensemble using log-loss optimization. We used the GradientBoostingClassifier from scikit-learn with n_estimators=200.

*LGBM* The LightGBM (LGBM) constitutes a more lightweight implementation of GBT. We used HistGradientBoostingRegressor for regression tasks and HistGradientBoostingClassifier for classification tasks from scikit-learn with the standard parameters.

### C.3 Hardware details

The experiments were mainly run on an computation cluster on hyperthreaded Intel Xeon E5-2697 v3 CPUs clocking at with 2.6Ghz. In total the experiments took around 300 CPU hours (30 CPUs for 10 h) on the cluster. This mainly stems from the number of parameters and different initializations. Before running the experiments on the cluster, the implementations were validated on a Dell XPS 15 9510 containing an Intel i7-11800 H at 2.30GHz. The laptop was running for around 12 h for the validation.

### C.4 Additional time complexity

As described in 4.1, the runtime of iPFI scales linearly with the number of features. This relationship is illustrated in Fig. 10. Each dataset or data stream was explained in ten independent iterations. The average explanation time in relation to the time without explaining was averaged and plotted over the feature count. A linear regression describes the relationship between $(0.104 \cdot |D|)$ the relative explanation time and feature count with an $R^2 = 0.966$ implying a linear effect.

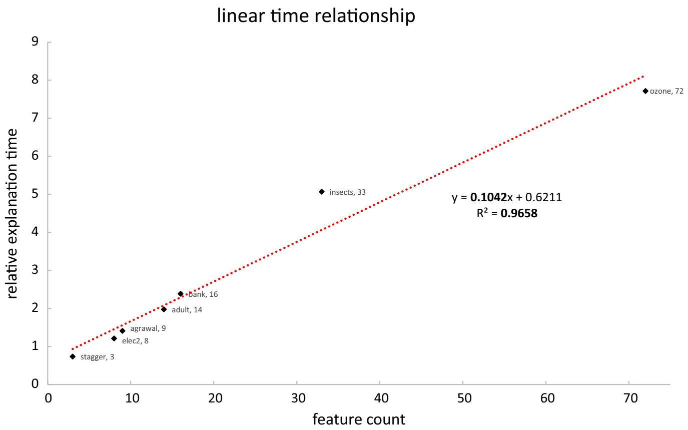


**Fig. 10** iPFI's runtime scales linearly with the feature count. The relative explanation time is averaged over 10 independent runs

### C.5 Summary of incremental experiments

Table 3 contains summary information about the supplementary experiments conducted in the incremental learning scenario (cf. Sect. 4.1). Figures 11, 12, 13, 14, 15, and 16 illustrate the experiments conducted on the *agrawal* concept drift data streams. Figure 17 shows our additional experiments conducted on the synthetic *stagger* concept drift data streams. Lastly, Fig. 18 shows the experiments conducted on the *elec2* data stream with an induced concept drift. The corresponding entries in Table 3 denote the approximation qualities for these experiments.

### C.6 Summary of batch experiments

Next to single batch experiment showcased in Sect. 4.3 and Fig. 9, we also show the results for the other datasets. Figures 19, 20, 21, 22 and 23 show the static batch model experiments for the other corresponding datasets.

## D Ground-truth PFI for the *agrawal* stream

River (Montiel et al., 2020) implements the *agrawal* (Agrawal et al., 1993) data stream with multiple classification functions. In our experiments we consider the following classification function (among others):

$$\begin{aligned}\text{Class A:}&((\text{age} < 40) \wedge (50K \leq \text{salary} \leq 100K)) \vee \\ &((40 \leq \text{age} < 60) \wedge (75K \leq \text{salary} \leq 125K)) \vee \\ &((\text{age} \geq 60) \wedge (25K \leq \text{salary} \leq 75K))\end{aligned}$$

Both feature *age* and *salary* are uniformly distributed with $X^{(\text{age})} \sim \mathcal{U}_{[20,80]}$ and $X^{(\text{salary})} \sim \mathcal{U}_{[20,150]}$. Given iid. samples from the data stream the classification problem can be transformed into a two-dimensional problem following the above defined classification

**Table 3** Summary of additional concept drift experiments on *agrawal*, *stagger*, and *elec2*

| Data stream | Sampling | $\alpha$ | Whole stream | | | | Before drift | | | | After drift | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Strategy | | $Q_2$ | IQR | $Q_1$ | $Q_3$ | $Q_2$ | IQR | $Q_1$ | $Q_3$ | $Q_2$ | IQR | $Q_1$ | $Q_3$ |
| agrawal fu. 1 | Uniform | .001 | .050 | .054 | .025 | .079 | .075 | .018 | .063 | .080 | .024 | .028 | .009 | .038 |
| | Geometric | .001 | .052 | .060 | .024 | .084 | .071 | .020 | .068 | .088 | .022 | .023 | .014 | .037 |
| | Uniform | .01 | .047 | .075 | .021 | .096 | .098 | .020 | .091 | .111 | .018 | .011 | .017 | .029 |
| | Geometric | .01 | .040 | .080 | .027 | .107 | .116 | .044 | .078 | .122 | .027 | .008 | .021 | .029 |
| agrawal fu. 2 | Uniform | .001 | .067 | .060 | .050 | .110 | .064 | .023 | .047 | .070 | .072 | .065 | .058 | .123 |
| | Geometric | .001 | .063 | .066 | .044 | .110 | .059 | .026 | .041 | .067 | .074 | .071 | .051 | .122 |
| | Uniform | .01 | .111 | .135 | .061 | .196 | .208 | .088 | .153 | .240 | .058 | .016 | .052 | .069 |
| | Geometric | .01 | .103 | .088 | .071 | .159 | .166 | .101 | .140 | .241 | .070 | .028 | .059 | .087 |
| agrawal fu. 2, early | Uniform | .001 | .067 | .123 | .035 | .158 | .110 | .060 | .080 | .140 | .064 | .105 | .032 | .137 |
| | Geometric | .001 | .066 | .132 | .036 | .168 | .116 | .067 | .082 | .149 | .063 | .106 | .032 | .138 |
| | Uniform | .01 | .069 | .113 | .052 | .165 | .217 | .043 | .195 | .238 | .066 | .042 | .046 | .088 |
| | Geometric | .01 | .078 | .103 | .055 | .157 | .187 | .020 | .177 | .196 | .069 | .042 | .050 | .092 |
| agrawal fu. 2, late | Uniform | .001 | .071 | .106 | .045 | .151 | .051 | .031 | .042 | .072 | .244 | .231 | .163 | .394 |
| | Geometric | .001 | .081 | .105 | .051 | .156 | .061 | .042 | .041 | .082 | .246 | .230 | .170 | .400 |
| | Uniform | .01 | .117 | .093 | .066 | .159 | .139 | .087 | .069 | .156 | .095 | .091 | .071 | .162 |
| | Geometric | .01 | .103 | .115 | .063 | .178 | .128 | .106 | .065 | .170 | .077 | .101 | .063 | .163 |
| agrawal fu. 3 | Uniform | .001 | .079 | .071 | .037 | .108 | .097 | .026 | .086 | .111 | .032 | .037 | .016 | .053 |
| | Geometric | .001 | .081 | .078 | .035 | .113 | .095 | .036 | .084 | .119 | .029 | .036 | .017 | .053 |
| | Uniform | .01 | .097 | .108 | .053 | .161 | .149 | .044 | .134 | .178 | .056 | .009 | .051 | .060 |
| | Geometric | .01 | .124 | .067 | .087 | .153 | .142 | .022 | .135 | .157 | .090 | .038 | .074 | .112 |
| agrawal fe. 1 | Uniform | .001 | .048 | .102 | .018 | .121 | .021 | .036 | .017 | .054 | .087 | .177 | .043 | .220 |
| | Geometric | .001 | .035 | .091 | .015 | .106 | .023 | .031 | .018 | .049 | .047 | .117 | .008 | .125 |
| | Uniform | .01 | .062 | .044 | .034 | .077 | .072 | .023 | .056 | .079 | .046 | .037 | .030 | .067 |
| | Geometric | .01 | .044 | .052 | .035 | .087 | .079 | .047 | .042 | .089 | .043 | .014 | .032 | .046 |
| stagger fu. 1 | Uniform | .001 | .018 | .118 | .014 | .132 | .009 | .005 | .007 | .012 | .132 | .443 | .075 | .518 |

**Table 3** (continued)

| Data stream | Sampling | $\alpha$ | Whole stream | | | | Before drift | | | | After drift | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Strategy | | $Q_2$ | IQR | $Q_1$ | $Q_3$ | $Q_2$ | IQR | $Q_1$ | $Q_3$ | $Q_2$ | IQR | $Q_1$ | $Q_3$ |
| | Geometric | .001 | .018 | .117 | .015 | .131 | .008 | .006 | .005 | .011 | .131 | .440 | .075 | .515 |
| stagger fe. 1 | Uniform | .001 | .270 | .305 | .042 | .347 | .041 | .061 | .033 | .093 | .353 | .068 | .311 | .378 |
| | Geometric | .001 | .037 | .039 | .033 | .072 | .037 | .066 | .032 | .098 | .037 | .022 | .036 | .057 |
| elec2 fe. 1, gradual | Uniform | .001 | .158 | .263 | .050 | .313 | .048 | .075 | .025 | .101 | .321 | .089 | .283 | .372 |
| | Geometric | .001 | .037 | .024 | .027 | .051 | .040 | .069 | .026 | .095 | .037 | .013 | .028 | .041 |

The image identifier point to the subsequent section of figures. $Q_2$ denotes the median of the error described in Experiment A computed for iPFI and interval PFI (solid line vs. dashed line in the figures). The interquartile range is calculated between $Q_1$ and $Q_3$

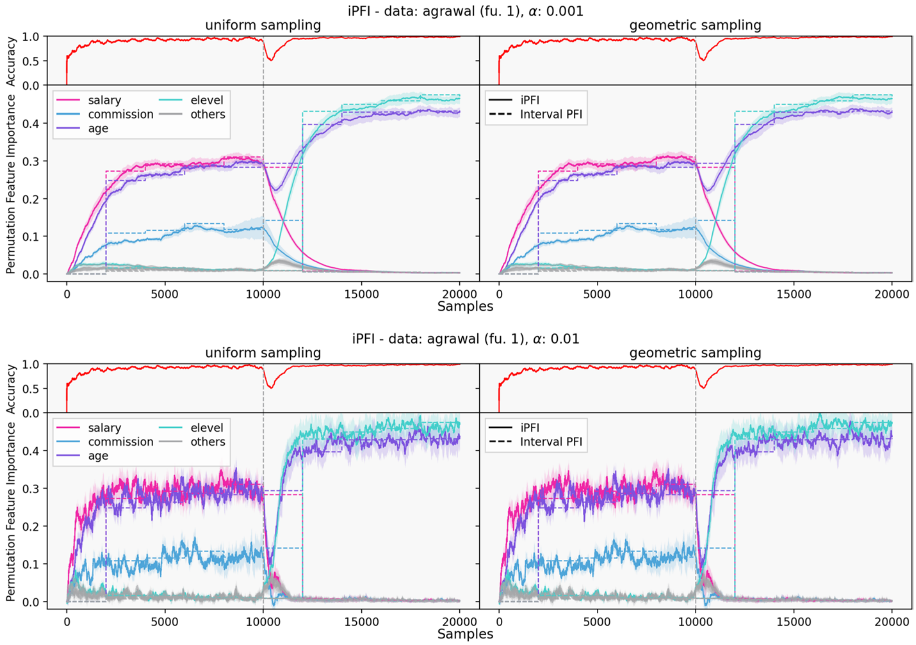


**Fig. 11** iPFI on *agrawal* with a function-drift (fu. 1) after 10k samples with $\alpha = 0.001$ (top) and $\alpha = 0.01$ (bottom)

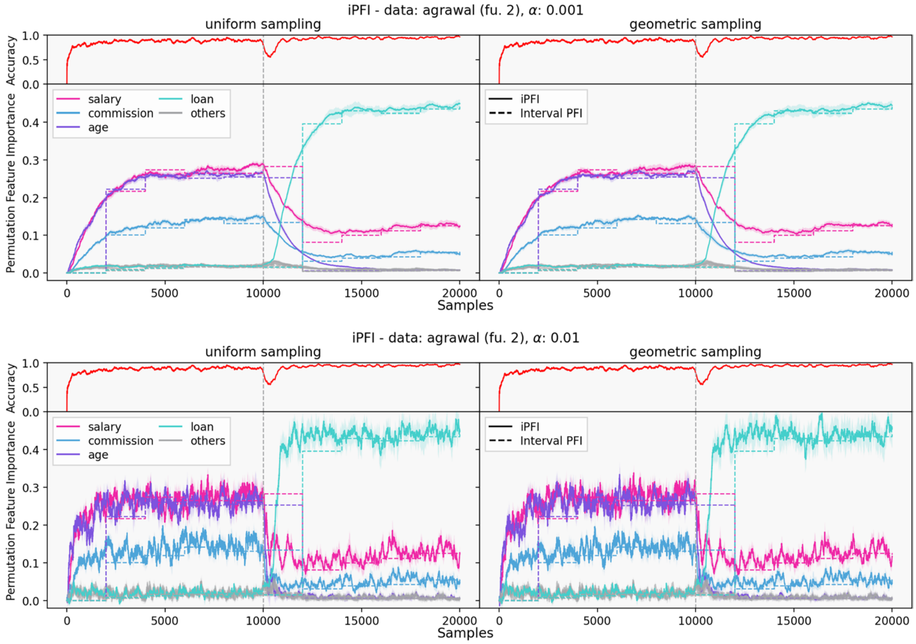


**Fig. 12** iPFI on *agrawal* with a function-drift (fu. 2) after 10k samples with $\alpha = 0.001$ (top) and $\alpha = 0.01$ (bottom)

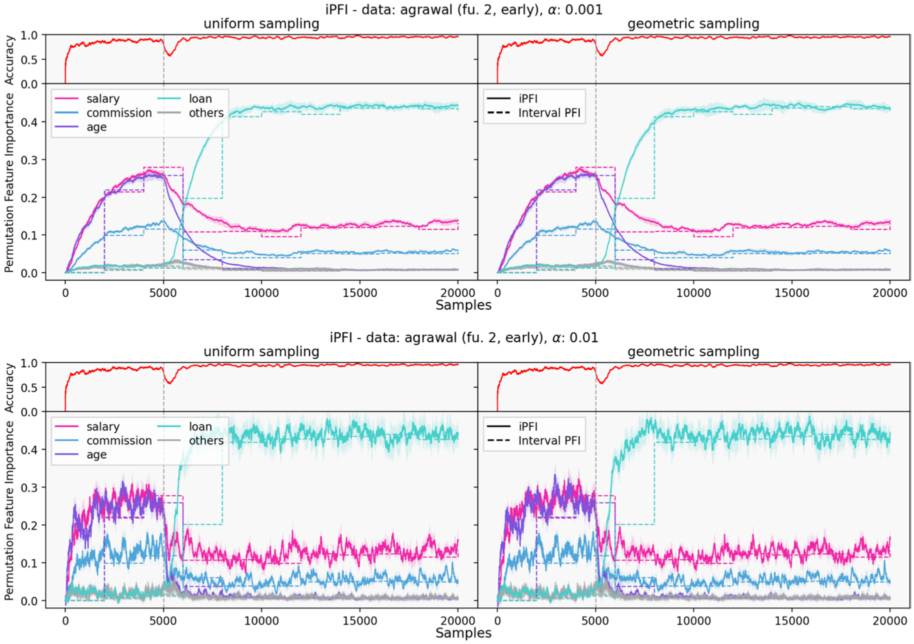


**Fig. 13** iPFI on *agrawal* with a function-drift (fu. 2, early) after 5k samples with $\alpha = 0.001$ (top) and $\alpha = 0.01$ (bottom)

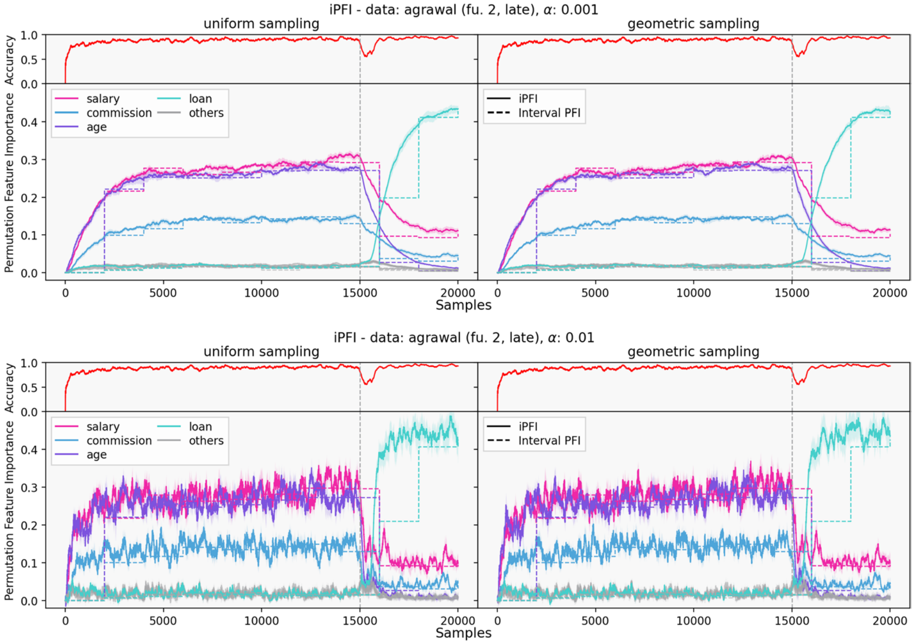


**Fig. 14** iPFI on *agrawal* with a function-drift (fu. 2, late) after 15k samples with $\alpha = 0.001$ (top) and $\alpha = 0.01$ (bottom)

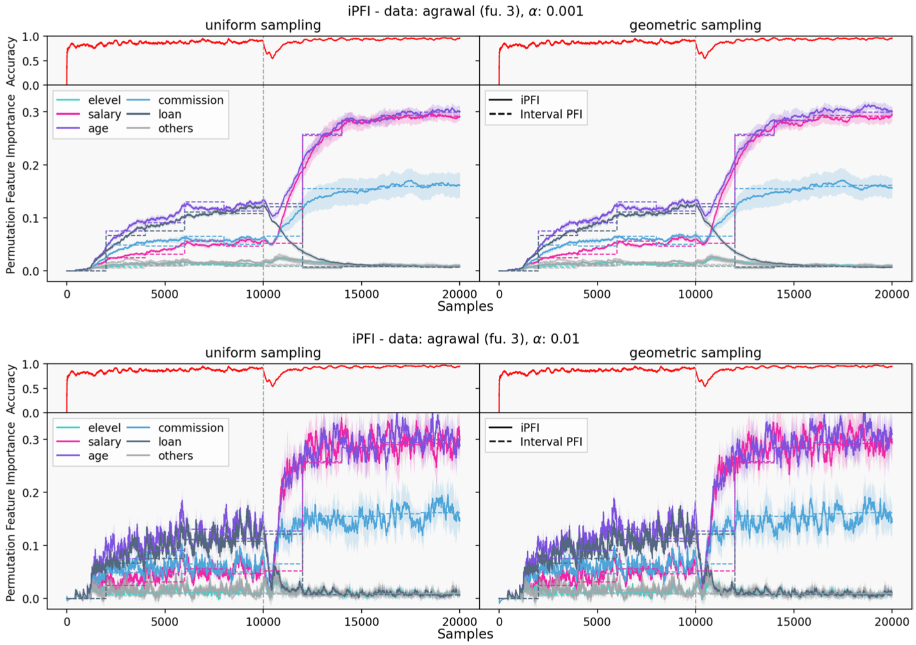


**Fig. 15** iPFI on *agrawal* with a function-drift (fu. 3) after 10k samples with $\alpha = 0.001$ (top) and $\alpha = 0.01$ (bottom)

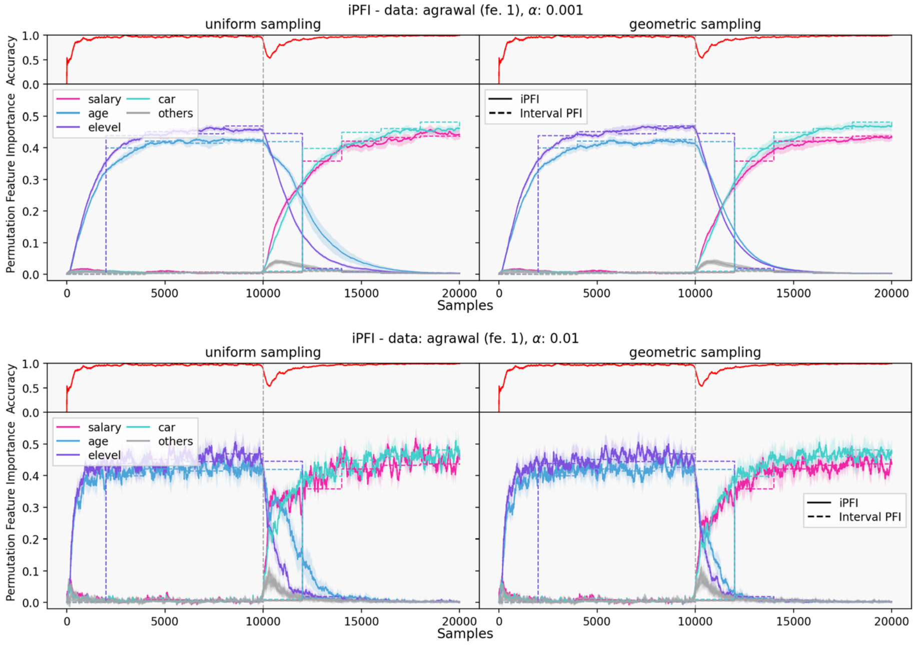


**Fig. 16** iPFI on *agrawal* with a feature-drift (fe. 1) after 10k samples with $\alpha = 0.001$ (top) and $\alpha = 0.01$ (bottom)

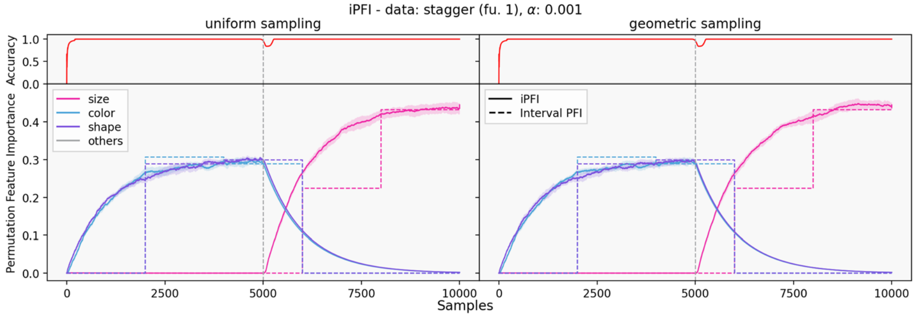


**Fig. 17** iPFI on *stagger* with a function-drift (fu. 1) after 5k samples with $\alpha = 0.001$

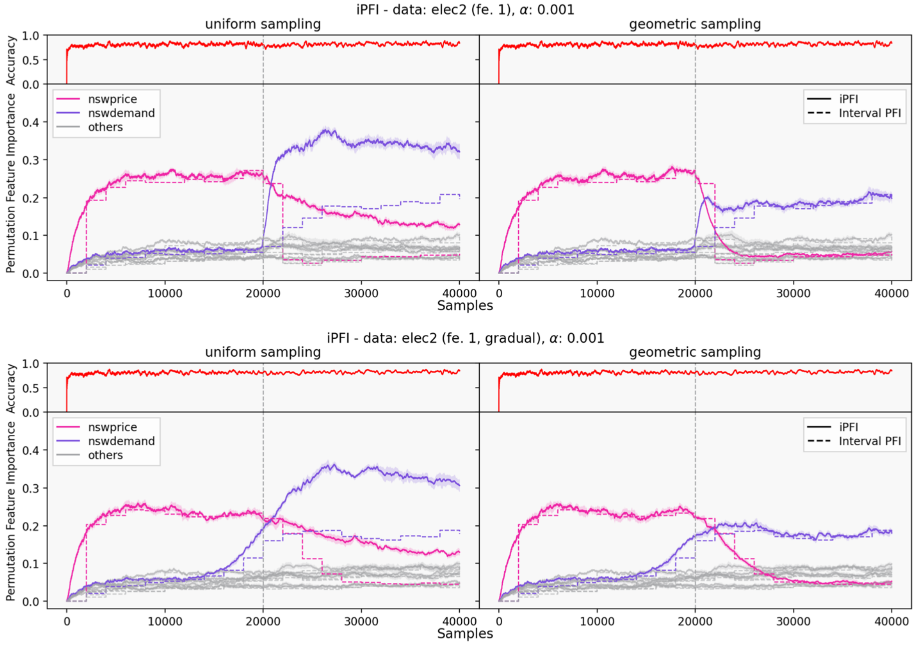


**Fig. 18** iPFI on *elec2* with a sudden feature-drift (fe. 1) (top) and a gradual feature-drift (fe. 1, gradual) (bottom) after 20k samples with $\alpha = 0.001$

function. The two-dimensional classification problem is illustrated in Fig. 24. A sample is classified as concept $A$ when it occurs contained in $A_1$, $A_2$, or $A_3$. Otherwise the sample is classified as concept $B$.

The theoretical PFIs can be calculated with the base probability of an sample belonging to concept A ($P(A_1) = P(A_2) = P(A_3) = \frac{5}{39}$) times the probability of switching the

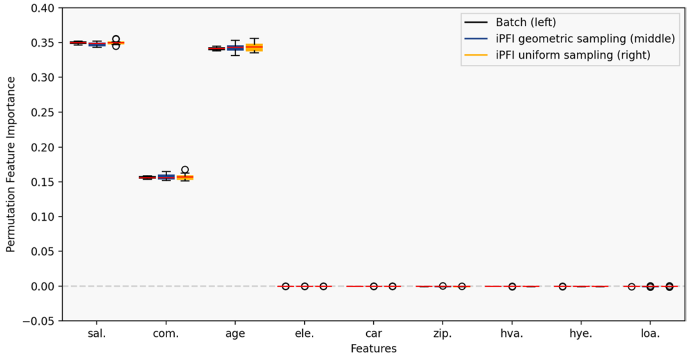


**Fig. 19** Boxplot of PFI estimates per feature of the *agrawal* dataset for batch baseline (left), iPFI with geometric sampling (middle), and iPFI with uniform sampling (right) on a pre-trained static LGBM

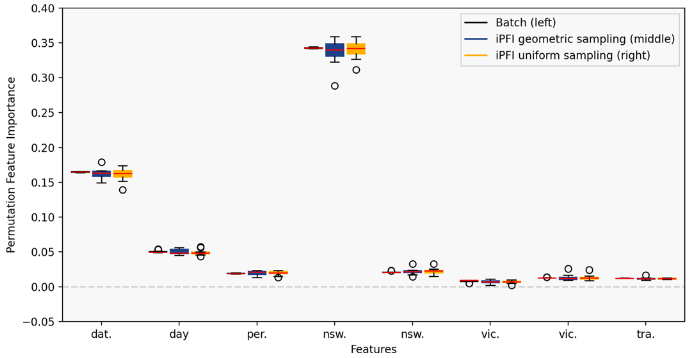


**Fig. 20** Boxplot of PFI estimates per feature of the *elec2* dataset for batch baseline (left), iPFI with geometric sampling (middle), and iPFI with uniform sampling (right) on a pre-trained static LGBM

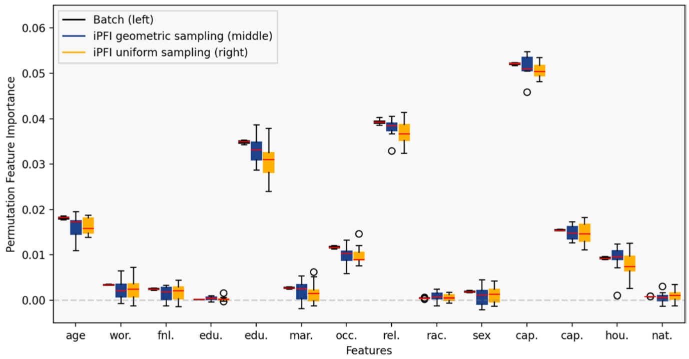


**Fig. 21** Boxplot of PFI estimates per feature of the *adult* dataset for batch baseline (left), iPFI with geometric sampling (middle), and iPFI with uniform sampling (right) on a pre-trained static GBT

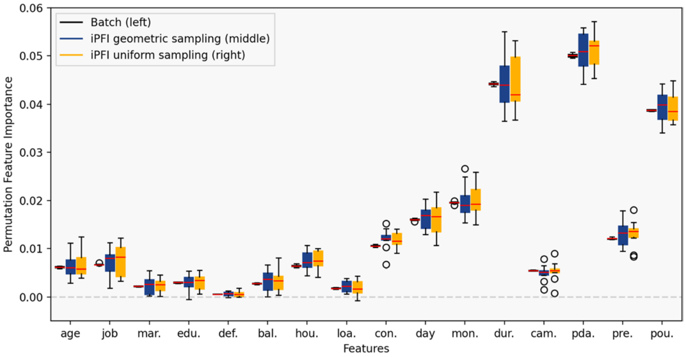


**Fig. 22** Boxplot of PFI estimates per feature of the *bank* dataset for batch baseline (left), iPFI with geometric sampling (middle), and iPFI with uniform sampling (right) on a pre-trained static NN

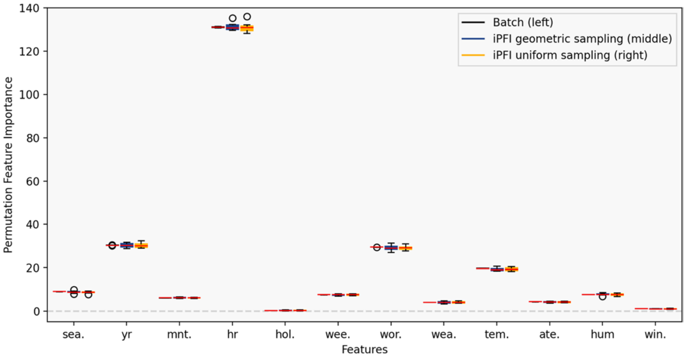
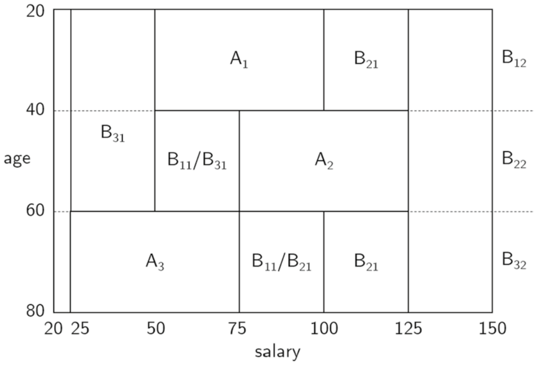


**Fig. 23** Boxplot of PFI estimates per feature of the *bike* dataset for batch baseline (left), iPFI with geometric sampling (middle), and iPFI with uniform sampling (right) on a pre-trained static LGBM

**Fig. 24** Two-dimensional classification problem of the *agrawal* data stream



class through changing a feature ($P(A_i \rightarrow B_{n,m})$) plus the vice versa for a sample originally belonging to concept $B$.

$$\begin{aligned}
\phi^{(\text{age})} &= P(A_1) \cdot P(A_1 \rightarrow B_{11}) + P(B_{11}) \cdot P(B_{11} \rightarrow A_1) \\
&\quad + P(A_2) \cdot P(A_2 \rightarrow B_{21}) + P(B_{21}) \cdot P(B_{21} \rightarrow A_2) \\
&\quad + P(A_3) \cdot P(A_3 \rightarrow B_{31}) + P(B_{31}) \cdot P(B_{31} \rightarrow A_3) = \\
&= \frac{5}{39} \cdot \frac{1}{3} + (\frac{5}{13} \cdot \frac{1}{3}) \cdot \frac{1}{3} \\
&\quad + 2 \cdot (\frac{5}{39} \cdot \frac{1}{2} + (\frac{5}{13} \cdot \frac{1}{3} + \frac{5}{13} \cdot \frac{1}{3} \cdot \frac{1}{2}) \cdot \frac{1}{3}) \approx \\
&\approx 0.3419 \\
\phi^{(\text{salary})} &= P(A_1) \cdot P(A_1 \rightarrow B_{12}) + P(B_{12}) \cdot P(B_{12} \rightarrow A_1) \\
&\quad + P(A_2) \cdot P(A_2 \rightarrow B_{22}) + P(B_{22}) \cdot P(B_{22} \rightarrow A_2) \\
&\quad + P(A_3) \cdot P(A_3 \rightarrow B_{32}) + P(B_{32}) \cdot P(B_{32} \rightarrow A_3) \\
&= 3 \cdot (\frac{5}{39} \cdot \frac{8}{13} + (\frac{8}{13} \cdot \frac{1}{3}) \cdot \frac{5}{13}) \approx \\
&\approx 0.4734
\end{aligned}$$

**Acknowledgements** We gratefully acknowledge funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation): TRR 318/1 2021-438445824.

**Author contributions** All authors contributed to the conception of the problem setting and overall design of the work. Fabian Fumagalli and Maximilian Muschalik elaborated on the technical details and main theoretical results, and drafted a first version of the manuscript. This version was revised and improved by all authors, who also read and approved the final manuscript.

**Funding** Open Access funding enabled and organized by Projekt DEAL. We gratefully acknowledge funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation): TRR 318/1 2021 - 438445824.

**Data availability** Not applicable, as all data sets used in this paper are publicly available or synthetically created. However, the mechanism for creating the data is described in detail in the paper.

**Code availability** The code is already publicly available at https://github.com/mmschlk/iPFI.

## Declarations

**Conflicts of interest** The authors have the following conflicts regarding the editorial board of the Machine Learning Journal and the journal track chair of the ECML-PKDD 2023: Willem Waegeman

**Ethics approval** The authors approve that this submission does not raise any potential ethical concerns.

**Consent to participate** All authors agreed with the content and all gave explicit consent to submit. Moreover, the authors obtained consent from the responsible authorities at the institute/organization where the work has been carried out, before the work was submitted. Finally, the authors consent that at least one author will participate at the ECML-PKDD 2023 in case of acceptance.

**Consent for publication** All authors consent to publish an individual's data or image, if this is required.

## References

Aas, K., Jullum, M., & Løland, A. (2021). Explaining individual predictions when features are dependent: More accurate approximations to Shapley values. *Artificial Intelligence, 298*, 103502. https://doi.org/10.1016/j.artint.2021.103502

Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access, 6*, 52138–52160. https://doi.org/10.1109/ACCESS.2018.2870052

Agrawal, R., Imielinski, T., & Swami, A. (1993). Database mining: A performance perspective. *IEEE Transactions on Knowledge and Data Engineering, 5*(6), 914–925. https://doi.org/10.1109/69.250074

Altmann, A., Toloşi, L., Sander, O., & Lengauer, T. (2010). Permutation importance: A corrected feature importance measure. *Bioinformatics, 26*(10), 1340–1347. https://doi.org/10.1093/bioinformatics/btq134

Archer, K. J., & Kimes, R. V. (2008). Empirical characterization of random forest variable importance measures. *Computational Statistics & Data Analysis, 52*(4), 2249–2260. https://doi.org/10.1016/j.csda.2007.08.015

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., & Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one, 10*(7), 0130140. https://doi.org/10.1371/journal.pone.0130140

Bahri, M., Bifet, A., Gama, J., Gomes, H. M., & Maniu, S. (2021). Data stream analysis: Foundations, major tasks and tools. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 11*(3), 1405. https://doi.org/10.1002/widm.1405

Barddal, J. P., Enembreck, F., Gomes, H. M., Bifet, A., & Pfahringer, B. (2019). Boosting decision stumps for dynamic feature selection on data streams. *Information Systems, 83*, 13–29. https://doi.org/10.1016/j.is.2019.02.003

Bifet, A., Gavaldà, R. (2009). Adaptive learning from evolving data streams. In *Advances in intelligent data analysis VIII, 8th international symposium on intelligent data analysis (IDA 2009)*, pp. 249–260 . https://doi.org/10.1007/978-3-642-03915-7_22.

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Calle, M. L., & Urrea, V. (2011). Letter to the editor: Stability of random forest importance measures. *Briefings in Bioinformatics, 12*(1), 86–89. https://doi.org/10.1093/bib/bbq011

Casalicchio, G., Molnar, C., Bischl, B. (2018). Visualizing the feature importance for black box models. In *Proceedings of machine learning and knowledge discovery in databases - European conference, (ECML PKDD 2018)*, pp. 655–670. https://doi.org/10.1007/978-3-030-10925-7_40

Cassidy, A. P., Deviney, F. A. (2014). Calculating feature importance in data streams with concept drift using online random forest. In *2014 IEEE international conference on big data (Big Data 2014)*, pp. 23–28 . https://doi.org/10.1109/BigData.2014.7004352.

Chen, H., Janizek, J. D., Lundberg, S. M., Lee, S. (2020). True to the model or true to the data? CoRR arXiv:abs/2006.16234

Covert, I., Lee, S.-I. (2021). Improving kernelshap: Practical shapley value estimation using linear regression. In *Proceedings of international conference on artificial intelligence and statistics (AISTATS 2021)*, pp. 3457–3465.

Covert, I., Lundberg, S. M., Lee, S. -I. (2020). Understanding global feature contributions with additive importance measures. In *Proceedings of international conference on neural information processing systems (NeurIPS 2020)*, pp. 17212–17223.

Covert, I., Lundberg, S., & Lee, S.-I. (2021). Explaining by removing: A unified framework for model explanation. *Journal of Machine Learning Research, 22*(209), 1–90.

de Souza, V. M. A., dos Reis, D. M., Maletzke, A. G., Batista, G. E. A. P. A. (2020). Challenges in benchmarking stream learning algorithms with real-world data. *Data Mining and Knowledge Discovery, 34*(6), 1805–1858 . https://doi.org/10.1007/s10618-020-00698-5.

Doshi-Velez, F., Kim, B. (2017). Towards a rigorous science of interpretable machine learning . https://arxiv.org/abs/1702.08608

Fanaee-T, H., & Gama, J. (2014). Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence, 2*(2), 113–127. https://doi.org/10.1007/s13748-013-0040-3

Feurer, M., van Rijn, J.N., Kadra, A., Gijsbers, P., Mallik, N., Ravi, S., Mueller, A., Vanschoren, J., Hutter, F. (2020). OpenML-python: An extensible python API for OpenML . https://arxiv.org/abs/1911.02490.

Fisher, A., Rudin, C., & Dominici, F. (2019). All models are wrong, but many are useful: Learning a Variable's importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research, 20*(177), 1–81.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics, 29*(5), 1189–1232. https://doi.org/10.1214/aos/1013203451

Frye, C., Mijolla, D. D., Begley, T., Cowton, L., Stanley, M., Feige, I. (2021). Shapley explainability on the data manifold. In *International conference on learning representations (ICLR 2021)*. https://openreview.net/forum?id=OPyWRrcjVQw.

García-Martín, E., Rodrigues, C. F., Riley, G., & Grahn, H. (2019). Estimation of energy consumption in machine learning. *Journal of Parallel and Distributed Computing, 134*, 75–88. https://doi.org/10.1016/j.jpdc.2019.07.007

Gomes, H. M., Mello, R. F. D., Pfahringer, B., Bifet, A. (2019). Feature scoring using tree-based ensembles for evolving data streams. In *2019 IEEE international conference on big data (Big Data 2019)*, pp. 761–769.

Gomes, H. M., Bifet, A., Read, J., Barddal, J. P., Enembreck, F., Pfharinger, B., Holmes, G., & Abdessalem, T. (2017). Adaptive random forests for evolving data stream classification. *Machine Learning, 106*(9), 1469–1495. https://doi.org/10.1007/s10994-017-5f642-8

Gregorutti, B., Michel, B., & Saint-Pierre, P. (2015). Grouped variable importance with random forests and application to multiple functional data analysis. *Computational Statistics & Data Analysis, 90*, 15–35. https://doi.org/10.1016/j.csda.2015.04.002

Gregorutti, B., Michel, B., & Saint-Pierre, P. (2017). Correlation and variable importance in random forests. *Statistics and Computing, 27*(3), 659–678. https://doi.org/10.1007/s11222-016-9646-1

Hapfelmeier, A., Hothorn, T., Ulm, K., & Strobl, C. (2014). A new variable importance measure for random forests with missing data. *Statistics and Computing, 24*(1), 21–34. https://doi.org/10.1007/s11222-012-9349-1

Harries, M. (1999). Splice-2 comparative evaluation: Electricity pricing. Technical report, The University of South Wales.

Haug, J., Braun, A., Zürn, S., Kasneci, G. (2022). Change detection for local explainability in evolving data streams. In *Proceedings of the 31st ACM international conference on information and knowledge management (CIKM 2022)*, pp. 706–716.

Hoeffding, W. (1948). A class of statistics with asymptotically normal distribution. *The Annals of Mathematical Statistics, 19*(3), 293–325. https://doi.org/10.1007/978-1-4612-0919-5_20

Hooker, G., Mentch, L., Zhou, S. (2019). Unrestricted permutation forces extrapolation: Variable importance requires at least one more model, or there is no free variable importance. https://arxiv.org/abs/1905.03151

Janzing, D., Minorics, L., Bloebaum, P. (2020). Feature relevance quantification in explainable AI: A causal problem. In *Proceedings of international conference on artificial intelligence and statistics (AISTATS 2020)*, pp. 2907–2916.

Jethani, N., Sudarshan, M., Covert, I. C., Lee, S.-I., Ranganath, R. (2021). Fastshap: Real-time shapley value estimation. In *Proceedings of international conference on learning representations (ICLR 2021)*.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. In *Proceedings of international conference on neural information processing system (NeurIPS 2017)*.

Kohavi, R. (1996). Scaling up the accuracy of Naive–Bayes classifiers: A decision-tree hybrid. In *Proceedings of international conference on knowledge discovery and data mining (KDD 1996)*, pp. 202–207.

König, G., Molnar, C., Bischl, B., Grosse-Wentrup, M. (2021). Relative feature importance. In *Proceedings of international conference on pattern recognition (ICPR 2021)*, pp. 9318–9325.

Losing, V., Hammer, B., & Wersing, H. (2018). Incremental on-line learning: A review and comparison of state of the art algorithms. *Neurocomputing, 275*, 1261–1274. https://doi.org/10.1016/j.neucom.2017.06.084

Lundberg, S. M., Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Proceedings of international conference on neural information processing systems (NeurIPS 2017)*, pp. 4768–4777.

Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., & Lee, S.-I. (2020). From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence, 2*(1), 56–67. https://doi.org/10.1038/s42256-019-0138-9

Molnar, C., König, G., Bischl, B., Casalicchio, G. (2020). Model-agnostic feature importance and effects with dependent features: A conditional subgroup approach.

Montiel, J., Halford, M., Mastelini, S. M., Bolmier, G., Sourty, R., Vaysse, R., Zouitine, A., Gomes, H. M., Read, J., Abdessalem, T., Bifet, A. (2020). River: Machine learning for streaming data in Python. https://arxiv.org/abs/2012.04740.

Moro, S., Laureano, R. M. S., Cortez, P. (2011). Using data mining for bank direct marketing: An application of the CRISP-DM methodology. In *Proceedings of the European simulation and modelling conference (ESM 2011)*.

Muschalik, M., Fumagalli, F., Hammer, B., & Hüllermeier, E. (2022). Agnostic explanation of model change based on feature importance. *KI - Künstliche Intelligenz*. https://doi.org/10.1007/s13218-022-00766-6

Nahmias, S., & Olsen, T. L. (2015). *Production and operations analysis*. Illinois: Waveland Press.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A. (2017). Automatic differentiation in pytorch. In *NIPS 2017 workshop on autodiff*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research, 12*, 2825–2830. https://doi.org/10.5555/1953048.2078195

Rényi, A. (1961). On measures of entropy and information. In *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability, volume 1: Contributions to the theory of statistics*, pp. 547–562.

Ribeiro, M. T., Singh, S., Guestrin, C. (2016). "Why Should I Trust You?": Explaining the predictions of any classifier. In *Proceedings of international conference on knowledge discovery and data mining (KDD 2016)*, pp. 1135–1144.

Schlimmer, J. C., & Granger, R. H. (1986). Incremental learning from noisy data. *Machine Learning, 1*(3), 317–354. https://doi.org/10.1007/BF00116895

Schlimmer, J. C., & Granger, R. H. (1986). Incremental learning from noisy data. *Machine Learning, 1*(3), 317–354. https://doi.org/10.1023/A:1022810614389

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision (ICCV 2017), pp. 618–626.

Shapley, L. S. (1953). A value for n-person games, volume II*Contributions to the Theory of Games (AM-28)* (pp. 307–318). New Jersey, USA: Princeton University Press.

Strobl, C., Boulesteix, A.-L., Zeileis, A., & Hothorn, T. (2007). Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics, 8*(1), 25. https://doi.org/10.1186/1471-2105-8-25

Strobl, C., Boulesteix, A.-L., Kneib, T., Augustin, T., & Zeileis, A. (2008). Conditional variable importance for random forests. *BMC Bioinformatics, 9*(1), 307. https://doi.org/10.1186/1471-2105-9-307

Turkay, C., Pezzotti, N., Binnig, C., Strobelt, H., Hammer, B., Keim, D. A., Fekete, J.-D., Palpanas, T., Wang, Y., Rusu, F. (2018). Progressive data science: Potential and challenges . https://arxiv.org/abs/1812.08032

Vitter, J. S. (1985). Random sampling with a reservoir. *ACM Transactions on Mathematical Software, 11*(1), 37–57. https://doi.org/10.1016/j.ipl.2005.11.003

Wang, H., Yang, F., & Luo, Z. (2016). An experimental study of the intrinsic stability of random forest variable importance measures. *BMC Bioinformatics, 17*(1), 60.

Wastensteiner, J., Weiss, T. M., Haag, F., Hopf, K. (2021). Explainable AI for tailored electricity consumption feedback: An experimental evaluation of visualizations. In *European conference on information systems (ECIS 2021)*, vol. 55.

Yuan, L., Pfahringer, B., Barddal, J. P. (2018). Iterative subset selection for feature drifting data streams. In *Proceedings of the 33rd annual ACM symposium on applied computing*, pp. 510–517.

Zhu, R., Zeng, D., & Kosorok, M. R. (2015). Reinforcement learning trees. *Journal of the American Statistical Association, 110*(512), 1770–1784. https://doi.org/10.1080/01621459.2015.1036994