iSAGE: An Incremental Version of SAGE for Online Explanation on Data Streams

Maximilian Muschalik[1,†,✉], Fabian Fumagalli[2,†], Barbara Hammer[2], and Eyke Hüllermeier[1]

[1] LMU Munich, MCML Munich, Geschwister-Scholl-Platz 1, Munich, Germany
[2] Bielefeld University, CITEC, Inspiration 1, Bielefeld, Germany
[†] denotes equal contribution
✉ maximilian.muschalik@ifi.lmu.de

Abstract. Existing methods for explainable artificial intelligence (XAI), including popular feature importance measures such as SAGE, are mostly restricted to the batch learning scenario. However, machine learning is often applied in dynamic environments, where data arrives continuously and learning must be done in an online manner. Therefore, we propose iSAGE, a time- and memory-efficient incrementalization of SAGE, which is able to react to changes in the model as well as to drift in the data-generating process. We further provide efficient feature removal methods that break (interventional) and retain (observational) feature dependencies. Moreover, we formally analyze our explanation method to show that iSAGE adheres to similar theoretical properties as SAGE. Finally, we evaluate our approach in a thorough experimental analysis based on well-established data sets and data streams with concept drift.

1 Introduction

If machine learning is used for high-stake decision-making, e.g., in healthcare [53] or energy consumption analysis [23], models learned on data should be transparent and explainable. However, as the best performing models are often opaque in nature, this is typically not the case. The field of explainable artificial intelligence (XAI) addresses this problem by developing methods to uncover the inner working of black box models and to make the input-output relationships represented by such models more understandable [2]. Notably, this includes *global feature importance* (global FI) methods, which quantify the influence of individual input features on the model predictions, and seek to rank the features in terms of their importance.

So far, XAI has mainly focused on static learning scenarios, where a single model is learned from data in a batch mode. However, in modern machine learning applications such as online credit risk scoring for financial services [13], intrusion detection in networks [4], or sensor network analysis [5, 16], data is not static but coming in the form of a continuously evolving stream of data. In applications of that kind, online algorithms are needed for learning in an incremental mode, processing data in a sequential manner one by one. Incremental

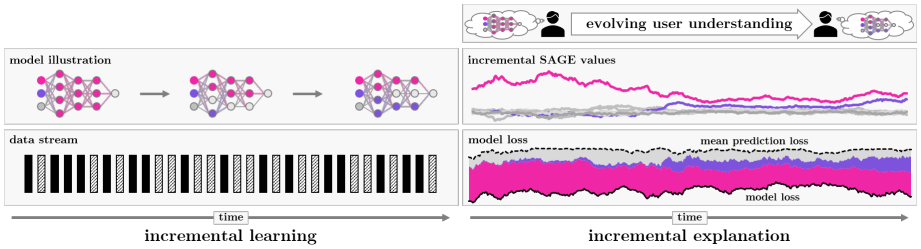

Fig. 1. An incremental model is fitted on a data stream. Incrementally explaining this model with iSAGE efficiently distributes the FI scores according to the model's loss evolving the user understanding of the model over time.

learning should not only be time- and memory-efficient, but must also account for possible changes in the underlying data distribution, which is referred to as *concept drift*. Such drift may occur in different forms and for different reasons, e.g., as a change of energy consumption patterns or hospital admission criteria due to pandemic-induced lockdowns [17].

In dynamic scenarios, where models are constantly evolving and reacting to their changing environment, static explanations do no longer suffice. Instead, explanations for monitoring dynamic models should be updated in a continuous manner, similar to the models themselves. In this work, we compute global FI in an incremental manner, thereby also addressing the challenge of drifting data distributions, where batch methods are likely to yield wrong explanations (cf. Fig. 7 in Appendix C). Providing an incremental global FI method comes with various challenges, not only conceptually and algorithmically, but also computationally, especially because the computation of many FI measures is already prohibitive in the batch setting.

Contribution. We take a first step towards efficient explanations for changing models on data streams and contribute:

- *iSAGE*; a model-agnostic global FI algorithm that provides time- and memory-efficient incremental estimates of SAGE values and is able to react to changes in the model and concept drift.
- *interventional and observational iSAGE*; two conceptual approaches to define SAGE values that extend on the existing discussion of appropriate feature removal techniques with an efficient incremental algorithm.
- *open source implementation*; a well-tested and general implementation of our algorithms and experiments that integrates into the well-known *River* [41] Python framework.[3]

Related Work. Global FI is an active part of XAI research, and various methods have been proposed [14]. Model-specific methods were developed based on

[3] iSAGE is implemented in `iXAI` at `https://github.com/mmschlk/iXAI`.

the magnitude of weights for linear models and neural networks [25, 30], as well as split heuristics for tree-based models [27]. Another common approach to global FI is to aggregate local explanations, such as model-agnostic LIME [47] and SHAP [39] or neural network specific methods [52, 57, 50, 51, 7]. Permutation Feature Importance (PFI) [8] is a well-established model-agnostic, global FI method with various extensions [40, 9, 36]. SAGE is based on the Shapley value [49], similar to SHAP [39] and LossSHAP [38] and overcomes computational limitations of aggregating local SHAP explanations. Retricting a model to compute FI is done either by retaining (*observational*) or breaking (*interventional*) feature dependencies, where it was shown that both methods generate different explanations and the choice should depend on the application [21, 1, 12].

Traditionally, XAI focuses on the batch learning scenario. However, recently more methods that natively support incremental, dynamic learning environments are proposed. For instance, online feature selection methods compute FI periodically [6, 56]. Haug et al. [28] propose a concept drift detection algorithm based on clusterings and changes in SHAP's base value. A model-specific approach for tree-based models is measuring the mean decrease in impurity (MDI) [10, 24]. In the notion of explaining change [43], iPFI [22] is a related model-agnostic approach that computes the traditional PFI [8] in an incremental manner. To efficiently restrict the model [15], we rely on geometric sampling [22] (interventional) and a combination of the conditional subgroup approach [40] and the TreeSHAP methodology [38] (observational).

Existing online FI methods are either model-specific or interpretation of the resulting feature importance scores is unintuitive, emphasizing the need for incremental variants of Shapley-based explanations, such as SAGE.

2 Shapley Additive Global Importance (SAGE)

Many feature importance techniques have been proposed in recent years [15], where each method allows to assess an importance ranking of the features. However, interpreting the exact scores and quantifying the difference between the importance of features remains unintuitive in many cases. Shapley-based explanations have attracted a lot of attention due to their unique mathematical properties, in particular the efficiency condition that ensures that the sum of these values over all features equals a specified model property, referred to as *model behavior* [15]. SHapley Additive Global Importance (SAGE) [14] is a well-known Shapley-based explanation technique that quantifies global FI as the contribution of individual features to the model's loss. SAGE is further a *model-agnostic* method that only relies on model evaluations and does not make any assumption about the inherent structure. In the following, we distinguish between the SAGE values ϕ, a statistical concept to define Shapley-based global FI, and the SAGE estimator $\hat{\phi}^{\text{SAGE}}$, an efficient approximator of the SAGE values. For a model $f : \mathcal{X} \to \mathcal{Y}$, the SAGE values $\phi(i)$ for every feature $i \in D$ are constructed, such that the sum is equal to the expected improvement in loss over using the mean

prediction $\bar{y} := \mathbb{E}_X[f(X)]$, i.e.

$$\nu(D) := \underbrace{\mathbb{E}_Y\left[\ell(\bar{y}, Y)\right]}_{\text{no feature information}} - \underbrace{\mathbb{E}_{(X,Y)}\left[\ell(f(X), Y)\right]}_{\text{with feature information}} = \sum_{i \in D} \phi(i),$$

where ℓ is a suitable loss (e.g, cross-entropy for classification, absolute error for regression, or kendall tau for rankings) and (X, Y) refers to the joint distribution of the data-generating random variables X and Y. The quantity $\nu(D)$ is viewed as the improvement in loss, if *all features* D are known to the model. It is then also natural to define $\nu(\emptyset) = 0$, i.e. the improvement in loss is expected to be zero, if *no features* are known to the model. To quantify the importance of single features, the expected improvement in loss, if only a subset $S \subset D$ of features is known, is introduced. To restrict this loss, the model is restricted to a subset of features $S \subset D$, by randomizing the features in $D \setminus S$. In the following, we write $f(x) = f(x^{(S)}, x^{(\bar{S})})$ to distinguish the features of x in S, $x^{(S)}$, and the features of x in $\bar{S} := D \setminus S$, $x^{(\bar{S})}$. To randomize the features in $\bar{S}$, we introduce the notation $f(x, S)$ with a set $S \subset D$ and the *observational* approach [39, 14]

$$f^{\text{obs}}(x, S) := \mathbb{E}\left[f(x^{(S)}, X^{(\bar{S})}) \mid X^{(S)} = x^{(S)}\right]$$

and the *interventional* approach [12, 32]

$$f^{\text{int}}(x, S) := \mathbb{E}\left[f(x^{(S)}, X^{(\bar{S})})\right].$$

The essential difference between the two approaches is that f^{int} breaks the dependence between the features in S and $\bar{S}$. The observational and interventional approach are also referred to as *on-manifold* and *off-manifold* explanation [21], or *conditional* and *marginal* expectation [32], respectively. While f^{int} is easy to approximate using the marginal distribution of the observed data points, approaches using f^{obs} rely on further assumptions on the conditional distribution [39, 1]. SAGE values are introduced using the observational approach but the SAGE algorithm relies on the interventional approach for approximation, i.e. assuming feature independence [14]. It was shown that both approaches yield significantly different explanations, if features are correlated [12, 21, 32]. We thus propose an algorithm for each approach and leave the choice of explanation to the practitioner, as it was concluded that this choice depends on the application scenario [12]. We define the restricted improvement in loss as

$$\nu(S) := \mathbb{E}_Y[\ell(\bar{y}, Y)] - \mathbb{E}_{(X,Y)}\left[\ell(f(X, S), Y)\right] \text{ for } f \in \{f^{\text{int}}, f^{\text{obs}}\}.$$

Then, $\nu : \mathcal{P}(D) \to \mathbb{R}$ defines a function over the powerset $\mathcal{P}(D)$, known as set function in cooperative game theory. The SAGE values [14] are then defined as the Shapley value [49] of ν, i.e. the *fair* attribution of $\nu(D)$ to individual features given its axiomatic propoerties.

Definition 1 (SAGE values [14]). *The SAGE values are defined as*

$$\phi(i) := \sum_{S \subset D \setminus \{i\}} \frac{1}{d} \binom{d-1}{|S|}^{-1} \left[\nu(S \cup \{i\}) - \nu(S)\right].$$

We refer to the interventional *and* observational *SAGE values, if f^{int} and f^{obs} are used for f in ν, respectively.*

Due to the exponential complexity of the Shapley value, the SAGE estimator uses a Monte-Carlo approximation [11] based on the representation

$$\phi(i) = \frac{1}{d!} \sum_{\pi \in \mathfrak{S}_D} \nu(u_i^+(\pi)) - \nu(u_i^-(\pi)) = \mathbb{E}_{\pi \sim \text{unif}(\mathfrak{S}_D)}[\nu(u_i^+(\pi)) - \nu(u_i^-(\pi))],$$

where $\mathfrak{S}_D$ is the set of permutations over D and $u_i^+(\pi)$ and $u_i^-(\pi)$ refer to the set of indices preceding feature i in π, in- and exclusively i. Plugging in the definition of ν and using Monte-Carlo estimation, the SAGE estimator is constructed.

Definition 2 (SAGE Estimator [14]). *Given data points $(x_n, y_n)_{n=1,\dots,N}$ and permutations $(\pi)_{n=1,\dots,N} \sim unif(\mathfrak{S}_D)$ the SAGE estimator is defined as*

$$\hat{\phi}^{SAGE}(i) := \frac{1}{N} \sum_{n=1}^{N} \ell(\hat{f}(x_n, u_i^-(\pi_n)), y_n) - \ell(\hat{f}(x_n, u_i^+(\pi_n), y_n)),$$

with $\hat{f}(x, \emptyset) := \frac{1}{N} \sum_{n=1}^{N} f(x_n)$ and $\hat{f}(x, S) := \frac{1}{M} \sum_{m=1}^{M} f(x^{(S)}, \tilde{x}_m^{(\bar{S})})$ for $\emptyset \neq S \subset D$ with $\tilde{x}_m$ sampled uniformly from $x_1, \dots, x_N$.

The mean prediction $\hat{f}(x, \emptyset)$ thereby differs to ensure that the SAGE values sum to the improvement in loss. For each permutation π_n and observation (x_n, y_n), the SAGE estimator can be efficiently computed by iterating through the permutation and evaluating ν on the preceding elements [11, 14]. The permutation sampling approach ensures that the efficiency condition of the Shapley value is maintained and thus the SAGE estimates sum approximately to $\nu(D)$. In contrast to other global FI measures, where interpretation of the scores are unintuitive, SAGE yields a meaningful axiomatic interpretation.

3 Incremental Global Feature Importance

In the following, we consider a data stream, where at time t the observations $(x_0, y_0), \dots, (x_t, y_t)$ have been observed. On this data stream, a model f_t is incrementally learned over time by updating $f_t \to f_{t+1}$ using the observation (x_t, y_t). [5, 37] Our goal is to estimate the (time-dependent) SAGE values ϕ_t alongside the incremental learning process using minimal resources. In particular, in an online learning scenario, where the model is constantly adapting, huge changes in global FI scores can occur, as has been observed in Haug et al. [28] and Fumagalli et al. [22]. To guarantee the reliability of the learned models, it is crucial to understand these global FI scores over time. The main challenge in estimating the SAGE values in an online learning scenario is that the model f_t and the data-generating random variables (X_t, Y_t) change over time and access to observations to compute $\hat{f}_t$ is limited.

While the SAGE estimator provides efficient estimates of static SAGE values for a given dataset, it does not react properly to changes in the model or concept drift. In Appendix C, we show an example (Fig. 7) which illustrates that the SAGE estimator yields wrong importance scores if the underlying distribution or model is not static. Furthermore, computing the SAGE estimator repeatedly in an incremental setting on a data stream quickly becomes infeasible. As a remedy, we propose incremental SAGE (iSAGE), an incremental estimator, which reacts to changing distributions and is able to explain dynamic, time-dependant models. To compare iSAGE in an incremental learning setting, we first propose Sliding Window SAGE (SW-SAGE), a time-sensitive baseline estimator that repeatedly computes the SAGE estimator on a sliding window.

Sliding Window SAGE (SW-SAGE) A naive approach of approximating SAGE values in an incremental manner is through repeated calculation within a sliding window (SW), which we denote as *SW-SAGE*. Applying SW-SAGE, necessitates storing all historical observations (x_t, y_t) for the last w (window length) observations, and recomputing the SAGE estimator from scratch based on the most up-to-date model f_t. The main computational effort of SW-SAGE stems from evaluating the model f_t and thus scales linearly with the window length w. The size w of the window has a profound effect on the resulting SAGE estimates. Choosing a large value for w, may increase the quality of the estimated SAGE values (larger sample), but can also lead to wrong importance scores, since the window may contain outdated observations. Vice versa, a window size too small leads to a high variance.

3.1 Incremental SAGE (iSAGE)

The high computational effort and the inability to reuse past results, because of the dynamic nature of f_t, strictly limits SW-SAGE in many scenarios, further discussed in Section 4.1. As a result, we now propose a time- and memory-efficient variant of SW-SAGE, which we refer to as *incremental SAGE* (iSAGE). The iSAGE algorithm computes the (time-dependent) SAGE values ϕ_t at time t and is able to react to changes in the model and concept drift, while updating its estimates efficiently in an incremental fashion with minimal computational effort. At each time step, we observe a sample (x_t, y_t) from the data stream, and our goal is to update the estimate using the current model f_t. We sample $\pi_t \sim \text{unif}(\mathfrak{S}_D)$ to compute the marginal contribution for $i \in D$ as

$$\Delta_t(i) := \ell(\hat{f}_t(x_t, u_i^-(\pi_t)), y_t) - \ell(\hat{f}_t(x_t, u_i^+(\pi_t)), y_t),$$

where $\hat{f}_t(x, S)$ is a time-sensitive approximation of the restricted model, further discussed in Section 3.2. These computations are then averaged over time, which yields the iSAGE estimator, outlined in Algorithm 1.

Definition 3 (iSAGE). *The iSAGE estimator is recursively defined as*

$$\textit{iSAGE: } \hat{\phi}_t(i) = (1 - \alpha) \cdot \hat{\phi}_{t-1}(i) + \alpha \cdot \Delta_t(i),$$

Algorithm 1 Incremental SAGE (iSAGE)

Require: stream $\{x_t, y_t\}_{t=1}^{\infty}$, feature indices $D = \{1, \ldots, d\}$, model f_t, loss function ℓ, and inner samples m

Initialize $\hat{\phi}^1 \leftarrow 0, \hat{\phi}^2 \leftarrow 0, \ldots, \hat{\phi}^d \leftarrow 0$, and smoothed mean prediction $y_\emptyset \leftarrow 0$
for all $(x_t, y_t) \in$ stream **do**
  Sample π, a permutation of D
  $S \leftarrow \emptyset$
  $y_\emptyset \leftarrow (1 - \alpha) \cdot y_\emptyset + \alpha \cdot f(x_t)$ {Udpate mean prediction}
  lossPrev $\leftarrow \ell(y_\emptyset, y_t)$ {Compute mean prediction loss}
  for $j = 1$ to d **do** {Iterate over π}
    $S \leftarrow S \cup \{\pi[j]\}$
    $y \leftarrow 0$
    for $k = 1$ to m **do** {Marginalize prediction with S}
      Sample $x_k^{(\bar{S})} \sim \mathbb{Q}_t^{(x,S)}$ {interventional (Algorithm 2) or observational (Algorithm 3)}
      $y \leftarrow y + f_t(x_t^{(S)}, x_k^{(\bar{S})})$
    end for
    $\bar{y} \leftarrow \frac{y}{m}$
    loss $\leftarrow \ell(\bar{y}, y_t)$
    $\Delta \leftarrow$ lossPrev $-$ loss
    $\hat{\phi}^{\pi[j]} \leftarrow (1 - \alpha) \cdot \phi^{\pi[j]} + \alpha \cdot \Delta$
    lossPrev $\leftarrow$ loss
  end for
end for
return $\phi^1, \phi^2, \ldots, \phi^d$

where $\alpha > 0$ *and computation starts at* $0 < t_0 < t$ *with* $\hat{\phi}_{t_0}(i) := \Delta_{t_0}(i)$.

The iSAGE estimator thus approximates ϕ_t by exponentially smoothing previous SAGE estimates, as $\mathbb{E}[\Delta_t(i)] = \phi_t(i)$. In the static batch setting, the SAGE estimator computes the restricted model $f_t(x, S)$ by sampling uniformly from observations in the dataset. However, when f_t is incrementally updated in the data stream setting, access to previous observations is limited as observations are discarded after the incremental update of the model. Furthermore, the distribution of previous observations might change over time, so recently observed samples should be preferred. We thus present two sampling strategies to implement the observational and interventional approach in an incremental fashion.

3.2 Incremental Feature Removal Strategies

As mentioned in Section 2, SAGE is defined using the observational approach, which is then approximated by the interventional approach, i.e. sampling from the marginal distribution and assuming feature independence. Clearly, this constitutes a strong assumption that is rarely satisfied in practice. Instead, we sample from the marginal distribution to compute *interventional* iSAGE and propose a novel approach to compute *observational* iSAGE, by approximating the conditional distribution. This aligns with [12], where it is claimed that the choices of

feature removal is dependent on the application scenario. For both approaches we now provide a time- and memory-efficient incremental sampling approach by maintaining time-dependent reservoirs to estimate $f(x, S)$.

Definition 4 (Estimator for $f(x, S)$). *At time t, we define for $\emptyset \neq S \subset D$*

$$\hat{f}_t(x, S) := \frac{1}{M} \sum_{m=1}^{M} f_t(x^{(S)}, \tilde{x}_m^{(\bar{S})}) \text{ with } x_1, \ldots, x_M \sim \mathbb{Q}_t^{(x,S)},$$

where $\bar{S} := D \setminus S$ and $\mathbb{Q}_t^{(x,S)}$ is a sampling distribution over features in $\bar{S}$. Further, $\hat{f}_t(x, \emptyset) := (1 - \alpha) f_{t-1}(x, \emptyset) + \alpha f_t(x_t)$ and $\hat{f}_{t_0}(x, \emptyset) := f_{t_0}(x_{t_0})$.

The interventional approach breaks the feature dependency and thus $\mathbb{Q}_t^{(x,S)}$ does not depend on the location x, whereas for the observational $\mathbb{Q}_t^{(x,S)}$ does depend on both, the location x as well as the subset S. We now describe incremental sampling algorithms to sample from $\mathbb{Q}^{(x,S)}$ for either approach.

Interventional iSAGE The interventional approach in the incremental learning setting is defined as $f_t^{\text{int}}(x, S) := \mathbb{E}\left[f_t(x^{(S)}, X_t^{(\bar{S})})\right]$. The batch SAGE algorithm samples uniformly from all observations from the given dataset. In an incremental learning scenario, this approach has significant drawbacks. First, access to previous observations is limited, as storing observations may be infeasible for the whole data stream. Second, the distribution of X_t may change over time, and it is, thus, beneficial to favor *recent* observations over older data points. The geometric sampling strategy, proposed by Fumagalli et al. [22], accounts for both of these challenges. Geometric sampling maintains *one* reservoir of length L, that is updated at each time step with an incoming data point by uniformly replacing a data point from the reservoir. Then, at each time step, observations $\tilde{x}_m$ are uniformly chosen from the reservoir. The geometric sampling strategy (fully initialized at time step $L := t_0$) thus chooses a previous observation from time r at time s with probability $L^{-1}(1 - L^{-1})^{s-r-1}$ for $r \geq L$, which clearly favors more recent observations. The complete procedure is given in Algorithm 2. At any time t, geometric reservoir sampling requires a storage space of $\mathcal{O}(L)$ data points. It has been shown that the geometric sampling procedure is favorable in scenarios with concept drift compared to memory-efficient uniform sampling approaches, such as general reservoir sampling [22].

Observational iSAGE The interventional approach can generate unrealistic observations when features are highly correlated, resulting in out-of-distribution evaluations of the model. When understanding causal relationships, it might be inappropriate to evaluate the model outside the data manifold [12], and we thus propose an alternative approach that can incorporate feature dependence in the incremental sampling process. The observational approach in the incremental setting is defined as $f_t^{\text{obs}}(x, S) := \mathbb{E}\left[f_t(x^{(S)}, X_t^{(\bar{S})}) \mid X_t^{(S)} = x^{(S)}\right]$. While observing data points x_t, we train for every feature $i \in D$ an incremental decision

tree that aims at predicting $x_t^{(i)}$ given the remaining feature values $x_t^{(D\setminus\{i\})}$. We then traverse the incremental decision tree using the input x_t and maintain a reservoir of length L at each leaf node, using the geometric sampling strategy described above, i.e. uniformly replacing an observation in the leaf's reservoir. This yields a reservoir of length L at every leaf node of the incremental decision tree, where both, the decision tree as well as the reservoir change over time. We propose to use a Hoeffding Adaptive Tree (HAT), a popular incremental decision tree [31], to adaptively maintain the structure. The approach can be viewed as an incremental variant of the *conditional subgroup* approach [40]. Given a subset $S \subset D$ and an observation x_t, we obtain the values of $\tilde{x}_m^{(\bar{S})}$ separately for each feature $j \in \bar{S}$. Using x_t, we traverse the HAT and at every decision node that splits on a feature in $\bar{S}$, we randomly split according to the split ratio of previous observed inputs, a statistic that is inherently available for a HAT. From the reservoir at the resulting leaf node, we then uniformly sample values for $\tilde{x}_m^{(j)}$ and repeat this process for every feature $j \in \bar{S}$ until we obtain all values for $\tilde{x}_m^{(\bar{S})}$. This methodology parallels the TreeSHAP approach of traversing decision trees for absent features, referred to as path dependent TreeSHAP [38]. Notably, our approach allows to extend the conditional subgroup approach to an arbitrary feature subset $S \subset D$ while maintaining only *one* decision tree *per feature* and further extends the approach to an incremental setting. The observational approach via HAT has a space complexity of $\mathcal{O}(d \cdot T^R \cdot L)$ where R refers to the HATs' maximum tree depth, T is the maximum number of tree splits, and L is the size of the reservoir at each leaf node.

3.3 Approximation Guarantees for Static Environments

We presented iSAGE as a time- and memory-efficient algorithm to estimate SAGE values over time incrementally. In contrast to the SAGE estimator, iSAGE reacts to changes in the model as well as concept drift, which we demonstrate empirically in Section 4. Analyzing iSAGE theoretically in an incremental learning scenario would require strong assumptions on the data-generating random variables (X_t, Y_t) and the approximation quality of the learned model f_t, as the iid assumption in general is not fulfilled. Instead, we now show theoretically that iSAGE has similar properties as the SAGE estimator in a static learning environment. In the following, we assume that $f \equiv f_t$ is a constant model and $(X, Y) \equiv (X_t, Y_t)$ a stationary data generating process. We further assume that $\mathbb{Q}_t^{(x,S)}$ is the true marginal (interventional) or conditional (observational) distribution and that samples are drawn iid, similar to Covert et al. [14].

Theorem 1. *For iSAGE* $\hat{\phi}_t(i) \to \phi_t(i)$ *for* $M \to \infty$ *and* $t \to \infty$.

Theorem 1 shows that iSAGE converges to the SAGE values. Further, the variance is controlled by α.

Theorem 2. *The variance of iSAGE is controlled by* α*, i.e.* $\mathbb{V}[\hat{\phi}_t(i)] = \mathcal{O}(\alpha)$.

Lastly, we show that iSAGE does not differ much from the SAGE estimator.

Theorem 3. *Given the SAGE estimator $\hat{\phi}_t^{SAGE}(i)$ computed at time t over all previously observed data points, it holds for iSAGE with $M \to \infty$, $\alpha = \frac{1}{t}$ and every $\epsilon > (1-\alpha)^{t-t_0+1}$ that $\mathbb{P}\left(|\hat{\phi}_t(i) - \hat{\phi}_t^{SAGE}(i)| > \epsilon\right) = \mathcal{O}(\frac{1}{t})$.*

While iSAGE admits similar properties as the SAGE estimator in a static environment, we showcase in our experiments that iSAGE is able to efficiently react to model changes and concept drift in an incremental learning setting.

4 Experiments

We now utilize iSAGE in multiple experimental settings. In Section 4.1, we show how iSAGE can be efficiently applied in dynamic environments with concept drift. In Section 4.2, we construct a synthetic ground-truth scneario for a data stream with concept drift and show that iSAGE is able to efficiently recover the SAGE values. In Section 4.3, we illustrate the difference of interventional and observational iSAGE, which yield profoundly different explanations. In Section 4.4, we show that iSAGE leads to the same results as the SAGE estimator in a static environment validating our theoretical results. As our iSAGE explanation technique is inherently model-agnostic, we train and evaluate our method on different incremental and batch models. [4]

4.1 iSAGE in Dynamic Environments with Concept Drift

In this experiment, we demonstrate the explanatory capabilities of iSAGE in a dynamic learning scenario with concept drift. We illustrate how iSAGE uncovers hidden changes in black box incremental models applied in real-world incremental learning scenarios where models are updated with every new observation. We compare iSAGE with incremental permutation feature importance (iPFI) [22], which is up to our knowledge, the only model-agnostic explanation method that can be applied in an incremental learning setting. For additional experiments and a comparison with the mean decrease in impurity (MDI) for tree-based models [24], we refer to the supplement material (cf., D.3). Fig. 2 explains the incremental learning procedure for an ARF classifier on the *elec2* data stream. Both methods detect similar feature importance rankings with varying absolute values. In contrast to iPFI, iSAGE explanations sum to the time-dependent difference in model loss over the loss using the mean prediction, due to the efficiency axiom of the Shapley value, which naturally increases interpretability of the method. Both methods correctly reveal the hidden changes in the model induced by the concept drift in the well-studied *elec2* [26]. The concept drift, which

[4] All model implementations are based on *scikit-learn* [46], *River* [41], and *torch* [45]. The data sets and streams are retrieved from *OpenML* [19] and *River*. The data sets are described in detail in Appendix D.1. `https://github.com/mmschlk/iSAGE-An-Incremental-Version-of-SAGE-for-Online-Explanation-on-Data-Streams`.

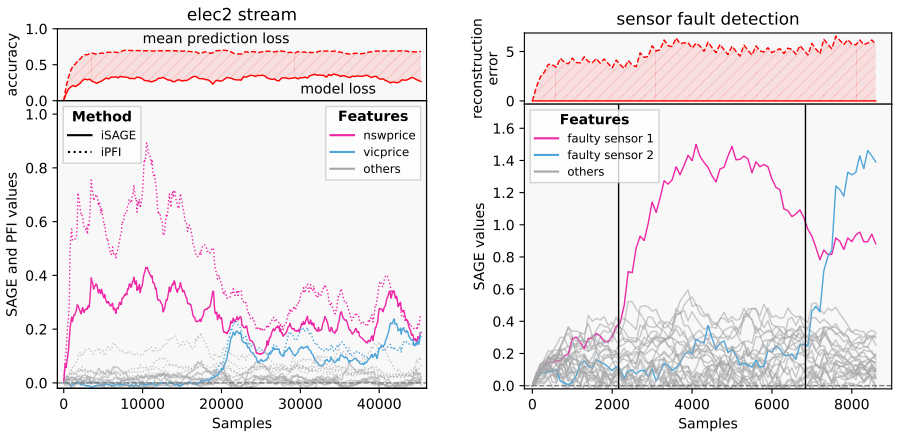

Fig. 2. iSAGE and iPFI of an ARF on *elec2* (left) and iSAGE for an incrementally fitted autoencoder for *fault detection* based on the reconstruction loss (right).

stems from the *vicprice* feature not having any values in the first $\approx 20k$ observations, would be obfuscated by solely plotting the model performance without any online explanations.

Localization of sensor faults. As an illustrative example, we conduct an experiment to show how online SAGE values can detect sensor faults in online sensor networks, which constitutes a challenging predictive maintenance problem [16, 54]. Similar to Hinder et al. [29], we simulate sensor network data of water pressures including sensor faults (vertical lines in Fig. 2 denote the time points) via the L-Town [55] simulation tool [34] and explain online learning models. We incrementally fit [5] and explain a NN autoencoder on the sensor readings. Fig. 2 shows how the autoencoder's reconstruction error is distributed onto the individual senor values by iSAGE. Notably, the faulty sensor can easily be identified through inspection of the iSAGE values after the sensor faulted.

4.2 Approximation quality with synthetic ground-truths.

We compare iSAGE to the inefficient baseline SW-SAGE, as well as synthetic ground-truth (GT) values estimated using the SAGE estimator. Conducting GT experiments in an incremental learning setting where models change with every new observation is computationally prohibitive. Moreover, it is not defined what constitutes a GT online explanation for real-world data streams with hidden drifts. We construct a data stream that consists of multiple sub-streams, each with different classification functions, i.e. inducing sudden concept drift when

[5] We fit the autoencoder with each new data point by conducting a single gradient update (i.e., batch size of 1). For further information about the experimental setup, we refer to Appendix D.3.

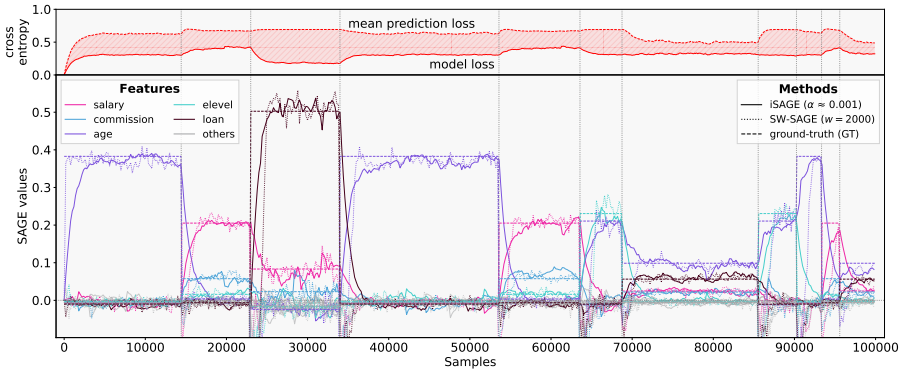

Fig. 3. iSAGE (solid), SW-SAGE (dotted) and GT (dashed) values for an example GT stream. SW-SAGE is computed with a stride of 100 ($0.05 \cdot w$) resulting in an overhead 20 times higher than iSAGE.

sub-streams are switched. Within each substream, we maintain a *static* pre-trained model with a pre-computed (constant) GT explanation. We observe how differently parameterized SW-SAGE and iSAGE estimators approximate the pre-computed GT values, see Fig. 3, and measure the approximation quality in terms of MSE and MAE. We repeat the complete experimental setup 20 times for each frequency scenario and summarize the resulting approximation errors (MSE) in Table 2 and Fig. 3. Independently of the substantially increased computational overhead (up to 20 times), SW-SAGE's approximation quality is substantially worse compared to iSAGE. In some scenarios, SW-SAGE reaches the GT values faster than iSAGE. Yet, in the important phases of change, SW-SAGE' estimates are substantially worse than iSAGE's (see Fig. 10 for a detailed view). This is a result from SW-SAGE attributing equal weight to outdated observations after a concept drift and the current model f_t classifying the samples

Table 1. Approximation quality of iSAGE (inc_c) and SW-SAGE (SW_c) on synthetic GT data streams for 20 iterations (c denotes the factor of additional model evaluations compared to iSAGE). The complete results are given in Table 2.

scenario		**high**		**middle**		**low**	
size (w)		500	1 000	500	1 000	500	1 000
MSE (σ)	inc_1	**0.034 (0.021)**	**0.038 (0.022)**	**0.027 (0.023)**	**0.027 (0.026)**	**0.015 (0.012)**	**0.013 (0.009)**
	SW_{20}	0.283 (0.262)	0.420 (0.360)	0.191 (0.271)	0.320 (0.487)	0.049 (0.043)	0.078 (0.081)
	SW_1	0.248 (0.198)	0.462 (0.413)	0.183 (0.200)	0.399 (0.792)	0.061 (0.067)	0.080 (0.079)

differently than the model before. iSAGE, however, smoothly changes between the different concepts.

4.3 Interventional and Observational iSAGE

In the presence of dependent variables, the choice of an interventional or observational approach has a profound effect on the SAGE values. In this experiment, we compare both approaches using the efficient incremental algorithms presented in Section 3.2. An ARF model is trained and explained on the synthetic *agrawal* data stream. The synthetic classification function is defined in Appendix D.3. In this stream the $X_{\text{commission}}$ feature ($X_{\text{com.}}$) directly depends on X_{salary}. Whenever the *salary* of an applicant exceeds $75k$, no *commission* is given ($X_{\text{com.}} = 0$), and otherwise the commission is uniformly distributed ($X_{\text{com.}} \sim U(10k, 75k)$). Fig. 4 showcases how interventional and observational iSAGE differ.
No significant importance is distributed to the $X_{\text{com.}}$ feature, if observational iSAGE is used, as the information present in $X_{\text{com.}}$ can be fully recovered by the observational approach based on X_{salary}. The importance is distributed onto the remaining two important features X_{salary} and X_{age}. However, when interventional iSAGE is used, the importance is also distributed to $X_{\text{com.}}$, as the model is evaluated outside the data manifold. The unrealistic feature values uncover that the incremental model has picked up on the transient relationship between the target values and the feature $X_{\text{com.}}$.

4.4 iSAGE and SAGE in Static Environments

We consider a static learning scenario, in which we compare interventional iSAGE with Covert et al. [14]'s original SAGE approach for well-established benchmark batch datasets. The models are pre-trained and then explained. We apply Gradient Boosting Trees (GBTs) [20], LightGBM models (LGBM) [33], and neural networks (NNs). The original SAGE explanations are directly computed from the batch datasets. iSAGE experiences the datasets as a randomly shuffled data stream where the model is not updated incrementally. We run this explanation procedure 20 times and illustrate the SAGE values on the *california* example dataset in Fig. 5 (more datasets in Section D.3). Fig. 5 shows that iSAGE approximates SAGE in the static setting on average with a higher variance. The higher variance is a direct result of the iSAGE having no access to future data points and the exponential smoothing mechanism controlled by α. iSAGE, thus, focuses more on recent samples, which is essential for non-stationary environments like incremental learning under concept drift.

5 Conclusion and Future Work

We propose and analyze iSAGE, a novel and model-agnostic explanation procedure to compute global FI in dynamic environments based on time-dependent SAGE values. In contrast to the batch SAGE algorithm [14], iSAGE is able to

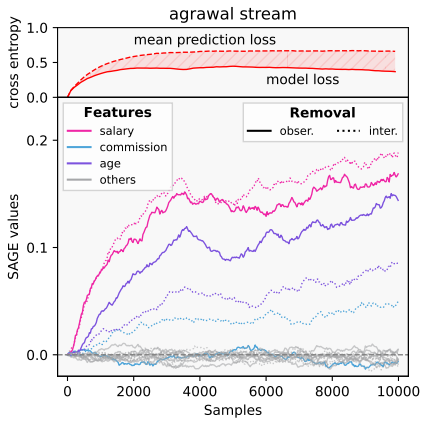

Fig. 4. Interventional and observational iSAGE for an ARF on an *agrawal* stream. The features have profoundly different scores.

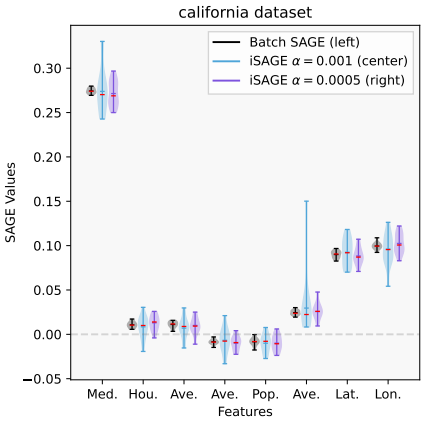

Fig. 5. SAGE values (median in red) per feature of the *california* dataset for the SAGE estimator (left), and interventional iSAGE (middle and right).

efficiently react to concept drift and changes in the model. We further extend SAGE with the observational and interventional SAGE values as distinctive objectives and present efficient incremental iSAGE variants, that are able to estimate these values over time and react to changes in the model and concept drift. In particular, we present an incremental approximation for the observational approach that combines the conditional subgroup approach [40] and the TreeSHAP methodology [38], which could also be used in a static learning environment to further improve the SAGE algorithm. We empirically confirm profound differences in both explanations depending on the choice of approach, which yields supporting arguments in the interventional and observational debate [21, 32, 12] that the choice should depend on the application scenario [12]. In a static environment, we prove that iSAGE has similar properties as SAGE and that both do not differ significantly. We further illustrate the efficacy of incremental explanations in multiple experiments on benchmark data sets and streams and conduct a ground-truth comparison.

Still, approximating Shapley values remains a computationally challenging problem. Moreover, this approach does not address the problem of incrementally decomposing the interactions between features, which requires further investigation. Finally, the interaction between human users and incrementally created explanations derived from methods like iSAGE need to be vigorously evaluated to identify further research opportunities.

Acknowledgements

We gratefully acknowledge funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation): TRR 318/1 2021 – 438445824. The authors want to thank Rohit Jagtani for supporting the implementation and valuable discussions. The authors want to thank Gunnar König for valuable discussions and feedback.

Bibliography

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

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

[3] Agrawal, R., Imielinski, T., Swami, A.: Database Mining: A Performance Perspective. IEEE Transactions on Knowledge and Data Engineering **5**(6), 914–925 (1993). https://doi.org/10.1109/69.250074

[4] Atli, B.G., Jung, A.: Online feature ranking for intrusion detection systems. CoRR **abs/1803.00530** (2018), `http://arxiv.org/abs/1803.00530`

[5] Bahri, M., Bifet, A., Gama, J., Gomes, H.M., Maniu, S.: Data stream analysis: Foundations, major tasks and tools. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **11**(3), e1405 (2021)

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

[7] Binder, A., Montavon, G., Lapuschkin, S., Müller, K.R., Samek, W.: Layer-wise relevance propagation for neural networks with local renormalization layers. In: Artificial Neural Networks and Machine Learning - 25th International Conference on Artificial Neural Networks ICANN 2016. Lecture Notes in Computer Science, vol. 9887, p. 63–71. Springer (2016). https://doi.org/10.1007/978-3-319-44781-0_8

[8] Breiman, L.: Random Forests. Machine Learning **45**(1), 5–32 (2001)

[9] Casalicchio, G., Molnar, C., Bischl, B.: Visualizing the Feature Importance for Black Box Models, Lecture Notes in Computer Science, vol. 11051, p. 655–670. Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-10925-7_40

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

[11] Castro, J., Gómez, D., Tejada, J.: Polynomial calculation of the shapley value based on sampling. Computers & Operations Research **36**(5), 1726–1730 (2009). https://doi.org/10.1016/j.cor.2008.04.004

[12] Chen, H., Janizek, J.D., Lundberg, S.M., Lee, S.: True to the model or true to the data? CoRR **abs/2006.16234** (2020), `https://arxiv.org/abs/2006.16234`

[13] Clements, J.M., Xu, D., Yousefi, N., Efimov, D.: Sequential deep learning for credit risk monitoring with tabular financial data. CoRR **abs/2012.15330** (2020), https://arxiv.org/abs/2012.15330

[14] Covert, I., Lundberg, S.M., Lee, S.: Understanding global feature contributions with additive importance measures. In: Advances in Neural Information Processing Systems 33: (NeurIPS 2020). pp. 17212–17223 (2020), https://proceedings.neurips.cc/paper/2020/hash/c7bf0b7c1a86d5eb3be2c722cf2cf746-Abstract.html

[15] Covert, I., Lundberg, S.M., Lee, S.I.: Explaining by Removing: A Unified Framework for Model Explanation. Journal of Machine Learning Research **22**(209), 1–90 (2021)

[16] Davari, N., Veloso, B., Ribeiro, R.P., Pereira, P.M., Gama, J.: Predictive maintenance based on anomaly detection using deep learning for air production unit in the railway industry. In: 8th IEEE International Conference on Data Science and Advanced Analytics (DSAA 2021). pp. 1–10. IEEE (2021). https://doi.org/10.1109/DSAA53316.2021.9564181

[17] Duckworth, C., Chmiel, F.P., Burns, D.K., Zlatev, Z.D., White, N.M., Daniels, T.W.V., Kiuber, M., Boniface, M.J.: Using explainable machine learning to characterise data drift and detect emergent health risks for emergency department admissions during covid-19. Scientific Reports **11**(1), 23017 (Dec 2021). https://doi.org/10.1038/s41598-021-02481-y

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

[19] Feurer, M., van Rijn, J.N., Kadra, A., Gijsbers, P., Mallik, N., Ravi, S., Müller, A., Vanschoren, J., Hutter, F.: Openml-python: an extensible python API for openml. Journal of Machine Learning Research **22**, 100:1–100:5 (2021), http://jmlr.org/papers/v22/19-920.html

[20] Friedman, J.H.: Greedy function approximation: A gradient boosting machine. The Annals of Statistics **29**(5), 1189–1232 (2001)

[21] Frye, C., Mijolla, D.d., Begley, T., Cowton, L., Stanley, M., Feige, I.: Shapley explainability on the data manifold. In: International Conference on Learning Representations (2021), https://openreview.net/forum?id=OPyWRrcjVQw

[22] Fumagalli, F., Muschalik, M., Hüllermeier, E., Hammer, B.: Incremental Permutation Feature Importance (iPFI): Towards Online Explanations on Data Streams. CoRR **abs/2209.01939** (2022), https://doi.org/10.48550/arXiv.2209.01939

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

[24] Gomes, H.M., Mello, R.F.d., Pfahringer, B., Bifet, A.: Feature scoring using tree-based ensembles for evolving data streams. In: 2019 IEEE International Conference on Big Data (Big Data 2019). p. 761–769 (2019)

[25] Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Mach. Learn. **46**(1–3), 389–422 (2002). https://doi.org/10.1023/A:1012487302797
[26] Harries, M.: Splice-2 comparative evaluation: Electricity pricing. Tech. rep., The University of South Wales (1999)
[27] Hastie, T., Tibshirani, R., Friedman, J.H.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Edition. Springer Series in Statistics, Springer (2009). https://doi.org/10.1007/978-0-387-84858-7
[28] Haug, J., Braun, A., Zürn, S., Kasneci, G.: Change detection for local explainability in evolving data streams. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management CIKIM 2022. pp. 706–716. ACM (2022). https://doi.org/10.1145/3511808.3557257
[29] Hinder, F., Vaquet, V., Brinkrolf, J., Hammer, B.: Model based explanations of concept drift. CoRR **abs/2303.09331** (2023), `https://doi.org/10.48550/arXiv.2303.09331`
[30] Horel, E., Mison, V., Xiong, T., Giesecke, K., Mangu, L.: Sensitivity based neural networks explanations. CoRR **abs/1812.01029** (2018), `http://arxiv.org/abs/1812.01029`, arXiv: 1812.01029
[31] Hulten, G., Spencer, L., Domingos, P.: Mining time-changing data streams. In: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining (KDD 2001). p. 97–106. ACM Press (2001). https://doi.org/10.1145/502512.502529
[32] Janzing, D., Minorics, L., Blöbaum, P.: Feature relevance quantification in explainable AI: A causal problem. In: The 23rd International Conference on Artificial Intelligence and Statistics (AISTATS 2020). Proceedings of Machine Learning Research, vol. 108, pp. 2907–2916. PMLR (2020)
[33] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.Y.: Lightgbm: A highly efficient gradient boosting decision tree. In: Advances in Neural Information Processing Systems 30 (NeurIPS 2017) (2017)
[34] Klise, K.A., Bynum, M., Moriarty, D., Murray, R.: A software framework for assessing the resilience of drinking water systems to disasters with an example earthquake case study. Environmental Modelling & Software **95**, 420–431 (2017). https://doi.org/https://doi.org/10.1016/j.envsoft.2017.06.022
[35] Kohavi, R.: 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). p. 202–207 (1996)
[36] König, G., Molnar, C., Bischl, B., Grosse-Wentrup, M.: Relative feature importance. In: Proceedings of International Conference on Pattern Recognition. pp. 9318–9325 (2021)
[37] Losing, V., Hammer, B., Wersing, H.: Incremental on-line learning: A review and comparison of state of the art algorithms. Neurocomputing **275**, 1261–1274 (2018). https://doi.org/10.1016/j.neucom.2017.06.084
[38] Lundberg, S.M., Erion, G., Chen, H., DeGrave, A., Prutkin, J.M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., Lee, S.I.: From local explanations to global understanding with explainable AI for trees. Nature Machine Intelligence **2**(1), 56–67 (2020)

[39] Lundberg, S.M., Lee, S.I.: A Unified Approach to Interpreting Model Predictions. In: Advances in Neural Information Processing Systems 30 (NeurIPS 2017). pp. 4768–4777 (2017)

[40] Molnar, C., König, G., Bischl, B., Casalicchio, G.: Model-agnostic feature importance and effects with dependent features - A conditional subgroup approach. CoRR **abs/2006.04628** (2020), https://arxiv.org/abs/2006.04628

[41] Montiel, J., Halford, M., Mastelini, S.M., Bolmier, G., Sourty, R., Vaysse, R., Zouitine, A., Gomes, H.M., Read, J., Abdessalem, T., Bifet, A.: River: machine learning for streaming data in python. J. Mach. Learn. Res. **22**, 110:1–110:8 (2021), http://jmlr.org/papers/v22/20-1380.html

[42] Moro, S., Cortez, P., Laureano, R.: 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) (2011)

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

[44] Nahmias, S., Olsen, T.L.: Production and operations analysis. Waveland Press, Illinois (2015)

[45] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch. In: Advances in Neural Information Processing Systems 30 (NeurIPS 2017) Workshop (2017)

[46] 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.: Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)

[47] Ribeiro, M.T., Singh, S., Guestrin, C.: "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In: Proceedings of International Conference on Knowledge Discovery and Data Mining, San Francisco. pp. 1135–1144 (2016)

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

[49] Shapley, L.S.: A Value for n-Person Games. In: Contributions to the Theory of Games (AM-28), Volume II, pp. 307–318. Princeton University Press (1953). https://doi.org/10.1515/9781400881970-018

[50] Shrikumar, A., Greenside, P., Kundaje, A.: Learning important features through propagating activation differences. In: Proceedings of the 34th International Conference on Machine Learning ICML 2017. Proceedings of Machine Learning Research, vol. 70, p. 3145–3153. PMLR (2017), http://proceedings.mlr.press/v70/shrikumar17a.html

[51] Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M.A.: Striving for simplicity: The all convolutional net. In: 3rd International Conference

on Learning Representations, ICLR 2015 (2015), http://arxiv.org/abs/1412.6806

[52] Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: Proceedings of the 34th International Conference on Machine Learning ICML 2017. Proceedings of Machine Learning Research, vol. 70, p. 3319–3328. PMLR (2017), http://proceedings.mlr.press/v70/sundararajan17a.html

[53] Ta, V.D., Liu, C.M., Nkabinde, G.W.: Big data stream computing in healthcare real-time analytics. In: Proceddings of International Conference on Cloud Computing and Big Data Analysis (ICCCBDA 2016). p. 37–42 (2016). https://doi.org/10.1109/ICCCBDA.2016.7529531

[54] Vaquet, V., Artelt, A., Brinkrolf, J., Hammer, B.: Taking care of our drinking water: Dealing with sensor faults in water distribution networks. In: Artificial Neural Networks and Machine Learning – (ICANN 2022. pp. 682–693. Springer Nature Switzerland, Cham (2022). https://doi.org/10.1007/978-3-031-15931-2_56

[55] Vrachimis, S., Eliades, D., Taormina, R., Kapelan, Z., Ostfeld, A., Liu, S., Kyriakou, M., Pavlou, P., Qiu, M., Polycarpou, M.: Battle of the leakage detection and isolation methods. Journal of Water Resources Planning and Management **148**, 04022068 (05 2022). https://doi.org/10.1061/(ASCE)WR.1943-5452.0001601

[56] Yuan, L., Pfahringer, B., Barddal, J.P.: Iterative subset selection for feature drifting data streams. In: Proceedings of the 33rd Annual ACM Symposium on Applied Computing. pp. 510–517 (2018)

[57] Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Computer Vision (ECCV 2014). Lecture Notes in Computer Science, vol. 8689, pp. 818–833. Springer (2014). https://doi.org/10.1007/978-3-319-10590-1_53

6 Ethical Statement

We propose iSAGE as a novel XAI method that *enables* explanations for any incrementally trained and dynamic black-box model. This is a novel research direction, which could lead to various use cases. Models, that could not be evaluated before, because of computational restrictions can be investigated with iSAGE. This enables high-performing models to be applied in various critical application domains such as healthcare [53], energy consumption analysis [23], credit risk scoring [13]. These application domains could greatly benefit from XAI methods such as iSAGE, since they can help in uncovering inherent biases or problems with fairness. This could help with more targeted regulation and scrutinization of opaque, yet high-performing, technologies than without explanations. On the other hand, improved interpretability may also lead to an increased acceptance and exploitation of potentially harmful applications using black box models.

Organization of the Supplement Material

The proofs of our Theorems are given in Section A. Section B contains technical details about the two feature removal strategies. Section C contains Covert et al. [14]'s original batch SAGE algorithm and illustrates the pitfalls of applying it in an dynamic learning environment. Finally, Section D contains additional information about the data sets and streams, models and experimental setup used in Section 4. Section D also includes further experimental results.

A Proofs

All proofs are based on the static environment assumption, i.e. $f_t \equiv f$, $(X_t, Y_t) \equiv (X, Y)$ and $\mathbb{Q}_t^{(x,S)}$ is the true marginal (interventional) or conditional (observational) distribution with samples are drawn iid, similar to Covert et al. [14].

A.1 Proof of Theorem 1

Proof. With $M \to \infty$ and sampling iid from the true marginal distribution for the interventional approach and the conditional distribution for the observational approach, the law of large numbers ensures that the approximation converges, i.e. $\hat{f}_t(x, S) \to f_t(x, S)$ for every t and $\emptyset \neq S \subset D$ almost surely. Furthermore, for $S = \emptyset$, the smoothed mean prediction fulfills

$$\mathbb{E}[\hat{f}_t(x, \emptyset)] = \alpha \sum_{s=t_0}^{t} (1-\alpha)^{t-s} \mathbb{E}[f_s(x_s)] \overset{t\to\infty}{\to} \mathbb{E}_X[f(X)].$$

Hence,

$$\begin{aligned}
&\mathbb{E}_{\pi_t}[\mathbb{E}_{(X,Y)}[\Delta_t(i)]] = \mathbb{E}_{\pi_t}[\mathbb{E}_{(X,Y)}[\ell(\hat{f}_t(x_t, u_i^-(\pi_t)), y_t) - \ell(\hat{f}_t(x_t, u_i^+(\pi_t)), y_t)]] \\
&\overset{M\to\infty}{\to} \mathbb{E}_{\pi_t}[\nu(u_i^-(\pi_t)) - \nu(u_i^+(\pi_t))] = \phi_t(i) = \phi(i),
\end{aligned}$$

as $f \equiv f_t$. Then, $\hat{\phi}_t(i)$ can be written as a weighted sum $\hat{\phi}_t(i) = \alpha \sum_{s=t_0}^{t} (1-\alpha)^{t-s} \Delta_s(i)$ and thus

$$\begin{aligned}
\mathbb{E}[\hat{\phi}_t(i)] &= \alpha \sum_{s=t_0}^{t} (1-\alpha)^{t-s} \mathbb{E}[\Delta_s(i)] \\
&\overset{M\to\infty}{\to} \alpha \sum_{s=t_0}^{t} (1-\alpha)^{t-s} \phi(i) = \phi(i)(1 - (1-\alpha)^{t-t_0+1}) \overset{t\to\infty}{\to} \phi(i).
\end{aligned}$$

A.2 Proof of Theorem 2

Proof. The variance of $\Delta_t(i)$ is constant and we denote $\sigma^2 := \mathbb{V}[\Delta_t(i)]$, where the variance is taken over the distribution of $(X, Y, \pi_t, \tilde{X}_1, \ldots, \tilde{X}_M)$, where $\pi_t \sim$

$\text{unif}(\mathfrak{S}_D)$ and $\tilde{X}_1, \ldots, \tilde{X}_M \sim \mathbb{Q}_t^{(x,S)}$. Furthermore, for two time steps s, t the random variables $\Delta_s(i), \Delta_t(i)$ are independent. Hence,

$$\mathbb{V}[\hat{\phi}_t(i)] = \alpha^2 \sum_{s=t_0}^{t} (1-\alpha)^{2(t-s)} \mathbb{V}[\Delta_t(i)] \leq \sigma^2 \frac{\alpha}{2-\alpha} = \mathcal{O}(\alpha),$$

where we have used the geometric series for $(1-\alpha)^2$ as an upper bound.

A.3 Proof of Theorem 3

Proof. It was shown in [14] that the SAGE estimator using t samples fulfills $\mathbb{V}[\hat{\phi}_t^{(\text{SAGE})}] = \mathcal{O}(\frac{1}{t})$ and thus

$$\begin{aligned}
\mathbb{P}\left(|\hat{\phi}_t(i) - \hat{\phi}_t^{\text{SAGE}}(i)| > \epsilon\right) &\leq \mathbb{P}\left(|\hat{\phi}_t(i) - \phi_t(i)| + |\phi_t(i) - \hat{\phi}_t^{\text{SAGE}}(i)| > \epsilon\right) \\
&\leq \mathbb{P}\left(|\hat{\phi}_t(i) - \phi_t(i)| > \epsilon\right) + \mathbb{P}\left(|\phi_t(i) - \hat{\phi}_t^{\text{SAGE}}(i)| > \epsilon\right) \\
&= \mathbb{P}\left(|\hat{\phi}_t(i) - \phi_t(i)| > \epsilon\right) + \mathcal{O}(\frac{1}{t}).
\end{aligned}$$

Now, for $M \to \infty$, the expectation of $\phi_t(i)$ is given as

$$\begin{aligned}
\mathbb{E}[\phi_t(i)] &= \alpha \sum_{s=t_0}^{t} (1-\alpha)^{t-s} \mathbb{E}[\Delta_s(i)] \\
&\overset{M\to\infty}{\to} \alpha \sum_{s=t_0}^{t} (1-\alpha)^{t-s} \phi(i) = \phi(i)(1 - (1-\alpha)^{t-t_0+1}).
\end{aligned}$$

Hence, by Chebyshev's inequality and Theorem 2

$$\begin{aligned}
\mathbb{P}\left(|\hat{\phi}_t(i) - \phi_t(i)| > \epsilon\right) &\leq \mathbb{P}\left(|\hat{\phi}_t(i) - \mathbb{E}[\hat{\phi}_t(i)]| + |\mathbb{E}[\hat{\phi}_t(i)] - \phi_t(i)| > \epsilon\right) \\
&= \mathbb{P}\left(|\hat{\phi}_t(i) - \mathbb{E}[\hat{\phi}_t(i)]| + |(1-\alpha)^{t-t_0+1}\phi(i)| > \epsilon\right) \\
&= \mathbb{P}\left(|\hat{\phi}_t(i) - \mathbb{E}[\hat{\phi}_t(i)]| > \epsilon - |(1-\alpha)^{t-t_0+1}\phi(i)|\right) \\
&\leq \mathcal{O}(\mathbb{V}[\hat{\phi}_t(i)]) = \mathcal{O}(\alpha) = \mathcal{O}(\frac{1}{t}),
\end{aligned}$$

where the assumption ensures that $\epsilon - |(1-\alpha)^{t-t_0+1}\phi(i)| > 0$. Finally,

$$\mathbb{P}\left(|\hat{\phi}_t(i) - \hat{\phi}_t^{\text{SAGE}}(i)| > \epsilon\right) = \mathcal{O}(\frac{1}{t}).$$

B Interventional and Observational Removal Strategies

We propose two distinct feature removal strategies for the *interventional* and the *observational* iSAGE as described in Section 2 and in particular Section 3.2. For the interventional approach, we propose to approximate the marginal feature distribution incrementally (see Appendix B.1). For the observational approach we propose to approximate the conditional data distribution (see Appendix B.2). Both approximation strategies can be efficiently computed and used to sample replacement values to restrict the model function for calculating the SAGE values.

Fig. 6, shows how both sampling approaches approximate the data distribution over time. The observational sampling approach (depicted with red crosses) clearly adheres to the dependencies in the data distribution, whereas the interventional approach breaks these dependencies. Also, in the setting without dependencies, both methods do not differ substantially as the observational approximation techniques reduces to approximating the interventional distribution.

B.1 Marginal (Interventional) Feature Removal with Reservoir Sampling

As described in Section 3.2, we apply geometric reservoir sampling to approximate a current estimate of the marginal feature distribution. The procedure in Algorithm 2 describes how new observations are stored in the reservoir. Sampling from the constructed reservoir is trivially defined by uniformly drawing a stored data point.

Algorithm 2 Updating the incremental geometric reservoir storage as described in [22]

Require: stream $\{x_t^D\}_{t=1}^{\infty}$ feature indices $D \leftarrow \{1, 2, \ldots, d\}$
Initialize reservoir $R \leftarrow \emptyset$ and number of seen samples with n
for all $x_t \in$ stream **do**
  $n \leftarrow n + 1$
  if $|R| \leq n$ **then**
    $R \leftarrow R \cup x_t$
  else
    $x_{\text{del}} \leftarrow \textsc{SampleUniformly}(R)$ {sample observation to remove from reservoir}
    $R \leftarrow (R \setminus \{x_{\text{del}}\}) \cup \{x_t\}$ {replace x_{del} with x_t}
  end if
end for

B.2 Conditional (Observational) Feature Removal with Incremental Decision Trees

As described in Section 3.2, we propose to train efficient, incremental decision tree models to more closely approximate the conditional data distribution. The procedure to train these incremental decision trees is given in Algorithm 3.

Algorithm 3 Updating the incremental trees storage mechanism for efficient conditional feature removal

Require: stream $\left\{x_t^D\right\}_{t=1}^{\infty}$ feature indices $D \leftarrow \{1, 2, \ldots, d\}$

Initialize the tree h_0^i, its leafs $L_0^i \leftarrow \emptyset$, the data reservoirs $R_0^i \leftarrow \emptyset$, and a leaf-reservoir mapping $M^i(.)$ for all $i \in D$
for all $x_t \in$ stream **do**
  for all $i \in D$ **do**
    $y_t^i \leftarrow x_t^i$
    $x^r \leftarrow x_t^{D \setminus \{i\}}$ {take rest of x as input features}
    $h_t^i \leftarrow$ LEARN_ONE $\left(h_{t-1}^i, x^r, y_t^i\right)$ {makes one incremental learning step with the remaining features as input}
    $L_t^i \leftarrow$ GET_LEAFS $\left(h_t^i\right)$ {traverses the tree and collects all leaf nodes}
    $R_t^i \leftarrow R_{t-1}^i \cup$ INITIALIZE $\left(M^i\left(L_t^i \setminus L_{t-1}^i\right)\right)$ {initialize new reservoirs at new leaf nodes}
    $R_t^i \leftarrow R_t^i \setminus M^i(L_{t-1}^i \setminus L_t^i)$ {delete outdated reservoirs}
    $l_t^i \leftarrow$ PREDICT_LEAF $\left(h_t^i, x^{D \setminus i}\right)$ {get the leaf node associated with a prediction given the remaining features}
    $r_{t-1}^i \leftarrow M^i(l_t^i)$ {get the reservoir associated with the leaf node}
    $r_t^i \leftarrow r_{t-1}^i \cup x_t$ {update the reservoir with current sample}
  end for
end for
return all trees h_t^i and reservoirs R_t^i for $i \in D$

Algorithm 4 Leaf Traversal Procedure

procedure TRAVERSE(node: n, sample: x, features present: S):
Initialize sampling ratios $W \leftarrow \emptyset$
if n is split node **then**
  if n splits on a feature present in S **then**
    $c \leftarrow \text{GETNEXTCHILD}(x)$ {make the split according to x}
    $n \leftarrow \text{TRAVERSE}(c, x)$
    return n
  end if
  $C \leftarrow \text{GETCHILDREN}(n)$ {get all children of n}
  for all nodes $c \in C$ **do**
    $W \leftarrow W \cup \text{GETWEIGHT}(c)$ {get weight of child in terms of how many samples have visited}
  end for
  $c \leftarrow \text{SAMPLEWITHWEIGHT}(C, W)$ {convert weights into probabilites and sample a child node accordingly}
  $n \leftarrow \text{TRAVERSE}(c, x)$
  return n
end if
return n {node n is a leaf node}

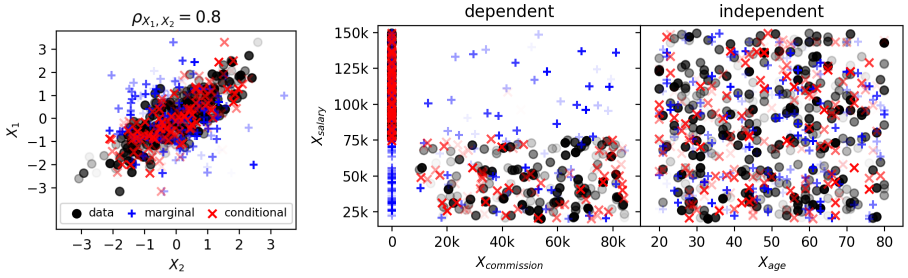

Fig. 6. Observational (red) and interventional (blue) feature removal strategies; The features follow the distributions $X_1 \sim \mathcal{N}(0,1)$, $X_2 \sim \mathcal{N}(0,1)$, $X_{\text{age}} \sim \text{unif}([20,80])$, $X_{\text{salary}} \sim \text{unif}([20k, 150k])$, and $X_{\text{commission}} = \mathbf{1}(X_{\text{salary}} \leq 75k) \cdot Q$ with $Q \sim \text{unif}([10k, 75k])$.

C The Batch SAGE Algorithm

Algorithm 5 contains the original sampling-based SAGE algorithm by Covert et al. [14]. The algorithm's notation is adjusted to fit into this paper's mathematical notation. Fig. 7 illustrates the pitfall of naively applying SAGE (as defined in Algorithm 5) in an incremental setting. The resulting importance values are incorrect because SAGE attributes equal weight onto each individual approximation step (in Lines 2 and 13 of Algorithm 5). Hence, older estimates that are no longer in-line with the real importance scores of a ever-changing model are given the same weight as more recent estimates.

Algorithm 5 Sampling-based approximation for SAGE values [14]

Require: data $\{x_i, y_i\}_{i=1}^N$, model f, loss function l, outer samples n, inner samples m

1: Initialize $\hat{\phi}^1 \leftarrow 0, \hat{\phi}^2 \leftarrow 0, \ldots, \hat{\phi}^d \leftarrow 0$
2: $y_\emptyset \leftarrow \frac{1}{N} \sum_{i=1}^N f(x_i)$ {Marginal Prediction}
3: **for all** $(x_i, y_i) \in$ data **do**
4: Sample π, a permutation of D
5: $S \leftarrow \emptyset$
6: lossPrev $\leftarrow \ell(y_\emptyset, y_i)$
7: **for** $j = 1$ to d **do**
8: $S \leftarrow S \cup \{\pi[j]\}$
9: $y \leftarrow 0$
10: **for** $k = 1$ to m **do**
11: Sample $x_k^{(\bar{S})} \sim \mathbb{Q}^{(x,S)}$ {In practice: $\mathbb{P}(X^{(\bar{S})})$}
12: $y \leftarrow y + f(x_i^{(S)}, x_k^{(\bar{S})})$
13: **end for**
14: $\bar{y} \leftarrow \frac{y}{m}$
15: loss $\leftarrow l(\bar{y}, y^i)$
16: $\Delta \leftarrow$ lossPrev $-$ loss
17: $\hat{\phi}^{\pi[j]} \leftarrow \phi^{\pi[j]} + \Delta$
18: lossPrev $\leftarrow$ loss
19: **end for**
20: **end for**
21: **return** $\frac{\phi^1}{n}, \frac{\phi^2}{n}, \ldots, \frac{\phi^d}{n}$

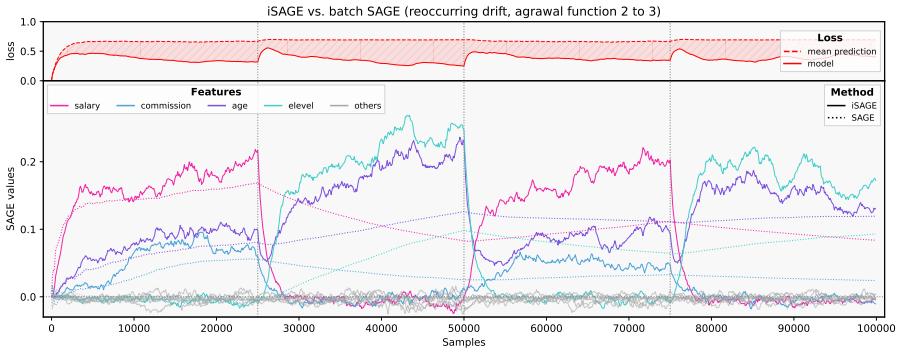

Fig. 7. Static SAGE in Dynamic Learning Environment: The original SAGE (dash-dotted) is calculated in a dynamic, incremental learning scenario. Classical SAGE yields false importance scores when it is applied for a changing model, as it gives each importance score equal weight. iSAGE (solid) with $\alpha = 0.001$, and $m = 5$ is provided as a reference.

D Experiments

This section contains additional information about the conducted experiments.

D.1 Data Set and Stream Descriptions

adult Binary classification dataset that classifies 48842 individuals based on 14 features into yearly salaries above and below 50k. There are six numerical features and eight nominal features. *adult* is a publicly available dataset [35].

bank Binary classification dataset that classifies 45211 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. *bank* is a publicly available dataset [42].

california Regression dataset containing 20640 samples of 8 numerical features with 1990 census information from the US state of California. The dataset is publicly available at `https://www.dcc.fc.up.pt/~ltorgo/Regression/cal_housing.html`

bike Regression dataset contains the hourly and daily count of rental bikes with information on weather. There are five numerical features and seven nominal features. *bike* is a publicly available dataset [18].

agrawal Synthetic data stream generator to create binary classification problems to decide whether an individual will be granted a loan based on nine features, six numerical and three nominal. There are ten different decision functions available. *agrawal* is a publicly available dataset [3].

stagger The *stagger* concepts makes a simple toy classification data stream. The synthetic 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. *stagger* is a publicly available dataset [48].

elec2 Binary classification dataset that classifies, if the electricity price will go up or down. The data was collected for 45312 time stamps from the Australian New South Wales Electricity Market and is based on eight features, six numerical and two nominal. The data stream contains a well-documented concept drift in its *vicprice* feature in that the feature has no values apart from zero in all observations up to $\approx 20\,000$ samples. After that the *vicprice* feature starts having values different from zero. *elec2* is a publicly available dataset [26].

L-Town Water Sensor Data L-Town is a popular variant of a virtual water distribution system, which is well-studied in the context of leakage detection algorithms [55]. Therein, sensor information can be simulated in different scenarios. We simulate pressure sensor readings over time and artificially introduce a sensor fault which needs to be detected. The simulation tool is publicly available [34].

D.2 Summary of the Incremental Explanation Procedure

Algorithm 6 illustrates the simplified explanation procedure. For each data point in a data stream, the incremental model first predicts the current's data point target label y_t from x_t. This label is used for prequential evaluation of the model's performance and to calculate the model's loss at time t. Then, the model is explained with this data point to update the the current iSAGE estimate $\hat{\phi}_t$. After the explanation is updated the learning procedure is triggered for an incremental learning step.

Algorithm 6 Incremental explanation procedure

Require: stream $\{x_t, y_t\}_{t=1}^{\infty}$, model $f(.)$, loss function $\mathcal{L}(.)$
1: **for all** $(x_t, y_t) \in$ stream **do**
2: $\quad \hat{y}_t \leftarrow f_t(x_t)$
3: $\quad \hat{\phi}_t \leftarrow$ `explain_one`(x_t, y_t)
4: $\quad f_{t+1} \leftarrow$ `learn_one`$(\mathcal{L}(\hat{y}_t, y_t))$
5: **end for**

D.3 Further Experimental Results

This subsection contains further results and details of the experiments conducted in Section 4. First, we present additional results for the stationary data setting. Second, we show examples of each synthetic GT scenario and provide further experimental results of different SW-SAGE window lengths and computational costs. Third, we show additional examples of iSAGE in real incremental learning scenarios from different data streams. Lastly, we provide the classification function to be learned omitted in Section 4.3.

Approximation Quality in Static Batch Learning Scenarios Fig. 8 contains SAGE values computed in a stationary data setting as described in 4.4. It shows the SAGE values for batch SAGE and iSAGE with $\alpha = 0.001$ and $\alpha = 0.0005$ for four datasets over mutliple runs. For the *bank* dataset, a LightGBM model was trained with *max_iterations* set to 70, a *learning_rate* of 0.2 and a *tree_depth* of 15. The LightGBM was explained in 20 independent runs. For the *california* dataset, an ARF regressor was trained in an incremental manner with *n_models* set to 3 and explained in 20 independent runs. For the *adult* dataset, an batch random forest classifier was trained with *n_models* set to 15 and explained in 10 independent runs. The *bike* dataset, a HAT regressor was trained in an incremental manner and then explained in 20 independent runs.

Approximation Quality of Synthetic ground-truth Data Streams In Fig. 9 we present an exemplary data stream for each of the three scenarios of the synthetic GT experiments conducted in Section 4.1. Fig. 9 shows how iSAGE and SW-SAGE approximate the pre-computed GT SAGE values for the pre-trained models. The runs in Fig. 9 also show how SW-SAGE has a higher approximation error in times of change, whereas iSAGE gradually and smoothly switches between the concepts.
For each scenario and iteration run, we pre-train incremental models, pre-compute the SAGE values in a batch mode, randomly shuffle the models in a synthetic data stream. For each run, we pre-train six individual ARF classifiers (an ensemble with 3 HATs) on data generators based on the first six *agrawal* concepts [3]. We train each ARF for 20 000 samples. After the pre-training, we compute the GT SAGE values according to Covert et al. [14]'s original SAGE definition. Therein, we apply feature-removal according to the marginal feature distribution with $m = 10$. Then, we create an artificial GT data stream by randomly switching between the different *agrawal* data generators yielding different data stream distributions. In each scenario, the probability of switching between the different pre-trained models is varied. For the setting with high-, middle, and low-frequency of changes, we set the probability of switching at each time (sample point) to $p_{\text{switch}} = 0.0005$, $p_{\text{switch}} = 0.0002$, and $p_{\text{switch}} = 0.0001$ respectively. On average, these probabilities result in a model change after 2 000, 5 000, and 10 000 samples for the high-, middle, and low-frequency scenarios.
In each scenario, we explain the underlying models with four different iSAGE

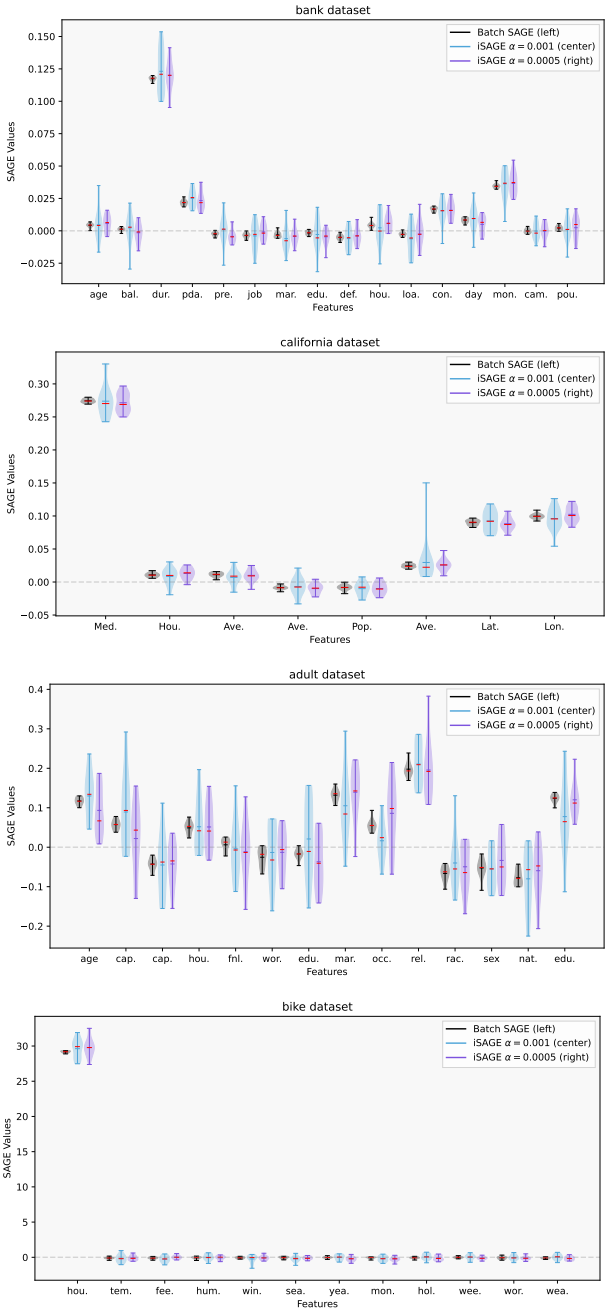

Fig. 8. SAGE values in stationary data setting calculated with batch SAGE and iSAGE for the *bank* (1st row), *california* (2nd row), *adult* (3rd row), and *bike* (4th row) data sets.

Table 2. Approximation quality of iSAGE and SW-SAGE on synthetic GT data streams for 20 iterations (inc_c denotes iSAGE and SW_c SW-SAGE with c denoting the factor of additional computational cost compared to iSAGE). (std. in brackets)

scenario		**low**				**middle**				**high**			
size (w)		100	500	1 000	2 000	100	500	1 000	2 000	100	500	1 000	2 000
MAE	inc_1	**.372** (.085)	**.197** (.052)	**.164** (.041)	**.153** (.040)	**.459** (.155)	**.254** (.080)	**.225** (.077)	**.226** (.074)	**.454** (.126)	**.284** (.075)	**.275** (.067)	**.305** (.072)
	SW_{20}	.384 (.092)	.227 (.057)	.219 (.060)	.261 (.085)	.492 (.172)	.335 (.111)	.361 (.121)	.480 (.175)	.505 (.139)	.452 (.130)	.575 (.176)	.811 (.249)
	SW_{10}	.384 (.092)	.228 (.057)	.220 (.061)	.263 (.084)	.492 (.172)	.336 (.111)	.365 (.124)	.487 (.180)	.506 (.139)	.456 (.131)	.580 (.176)	.820 (.254)
	SW_5	.384 (.092)	.230 (.058)	.221 (.061)	.264 (.081)	.494 (.173)	.338 (.112)	.371 (.130)	.494 (.184)	.508 (.140)	.462 (.132)	.592 (.184)	.839 (.257)
	SW_2	.385 (.093)	.233 (.061)	.235 (.066)	.285 (.092)	.494 (.171)	.349 (.113)	.389 (.133)	.534 (.208)	.512 (.142)	.482 (.137)	.615 (.190)	.879 (.274)
	SW_1	.386 (.092)	.244 (.064)	.246 (.070)	.306 (.098)	.495 (.171)	.364 (.121)	.426 (.160)	.566 (.216)	.519 (.143)	.506 (.148)	.670 (.214)	.921 (.345)
MSE	inc_1	**.057** (.042)	**.015** (.012)	**.013** (.009)	**.015** (.011)	**.103** (.105)	**.027** (.023)	**.027** (.026)	**.034** (.034)	**.094** (.066)	**.034** (.021)	**.038** (.022)	**.051** (.027)
	SW_{20}	.066 (.050)	.049 (.043)	.078 (.081)	.139 (.150)	.139 (.140)	.191 (.271)	.320 (.487)	.582 (.971)	.151 (.104)	.248 (.198)	.420 (.360)	.690 (.607)
	SW_{10}	.065 (.050)	.049 (.044)	.080 (.086)	.137 (.148)	.139 (.139)	.189 (.266)	.325 (.501)	.596 (.035)	.152 (.105)	.250 (.198)	.422 (.360)	.690 (.612)
	SW_5	.066 (.050)	.051 (.047)	.078 (.085)	.125 (.125)	.141 (.142)	.192 (.273)	.337 (.566)	.600 (.062)	.155 (.107)	.253 (.199)	.429 (.373)	.703 (.628)
	SW_2	.065 (.047)	.051 (.048)	.088 (.106)	.137 (.145)	.137 (.133)	.176 (.226)	.326 (.466)	.652 (.249)	.156 (.104)	.259 (.211)	.432 (.399)	.713 (.644)
	SW_1	.066 (.048)	.061 (.067)	.080 (.079)	.160 (.215)	.136 (.125)	.183 (.200)	.399 (.792)	.529 (.883)	.162 (.107)	.283 (.262)	.462 (.413)	.757 (.891)

and SW variants. The SW span $w_1 = 100$, $w_2 = 500$, $w_3 = 1\,000$, and $w_4 = 2\,000$ samples, and are computed with a stride (frequency) of $w_i/20$ samples. We couple iSAGE's α parameter with the window sizes $\alpha_i = 2/(w_i + 1)$ [44]. For both the SW-SAGE and iSAGE we set $m = 1$, the loss function to cross-entropy and apply the interventional feature removal approach.

Explaining Incremental Models Next to the experiments in Section 4.1, we compare iSAGE with related FI methods; namely, incremental permutation feature importance (iPFI) [22] and mean decrease in impurity (MDI) [24]. MDI, as a model-specific method, is only applicable on incremental decision trees. Hence, in Fig. 11, we compute the three methods for a HAT on the *elec2* [26] real world data stream as an example. All methods correctly identify the two most important features.

Online Sensor Fault Detection We conduct an experiment of online anomaly detection from sensor data. Therein, we simulate a data stream of sensor readings using the L-Town virtual water distribution network [55] with an open-source software [34] following Hinder et al. [29]. We create a data stream containing 29 pressure sensor readings (features). Two sensor fault are randomly introduced (one at $t = 2160$ and one at $t = 6840$). Fig. 13 shows the dataset of sensor readings. To illustrate how iSAGE can be used to explain any black-box model fitted on a data stream, we opted to fit an undercomplete autoencoder on the stream and explain its reconstruction error. The autoencoder is trained with the *river* and *deep-river* [6] open source framework for training NN models incrementally. The autoencoder consists of 4 layers. The encoder reduces the dimensionalty from the feature size to a hidden size of 10 and then to a latent dimension of 3. The decoder reverses this by expanding the latent dimension of 3 to a second hidden size of 10 and then back to the feature dimension. The input of the autoencoder is scaled with a standard scaler. The model is trained with a batch size of one (one model update with each new observation) with a relatively high learning rate of $\gamma = 0.05$. We explain this autoencoder with the intverventional iSAGE approach ($\alpha = 0.001$ and $m = 10$).

Interventional and Observational iSAGE Fig. 14 shows how the iSAGE values differ when we apply the observational feature removal mechanism compared to the interventional strategy. We conduct this experiment on an *agrawal* data stream with known feature dependencies and on the real-world data stream *elec2*. The classification function to be learned on the *agrawal* stream is defined as

$$\begin{aligned}\text{class 1: } &((X_{\text{age}} < 40) \wedge (50\,000 \leq X_{\text{salary}} \leq 100\,000)) \vee \\ &((40 \leq X_{\text{age}} < 60) \wedge (75\,000 \leq X_{\text{salary}} \leq 125\,000)) \vee \\ &((X_{\text{age}} \geq 60) \wedge (25\,000 \leq X_{\text{salary}} \leq 75\,000)).\end{aligned}$$

[6] `https://github.com/online-ml/deep-river`

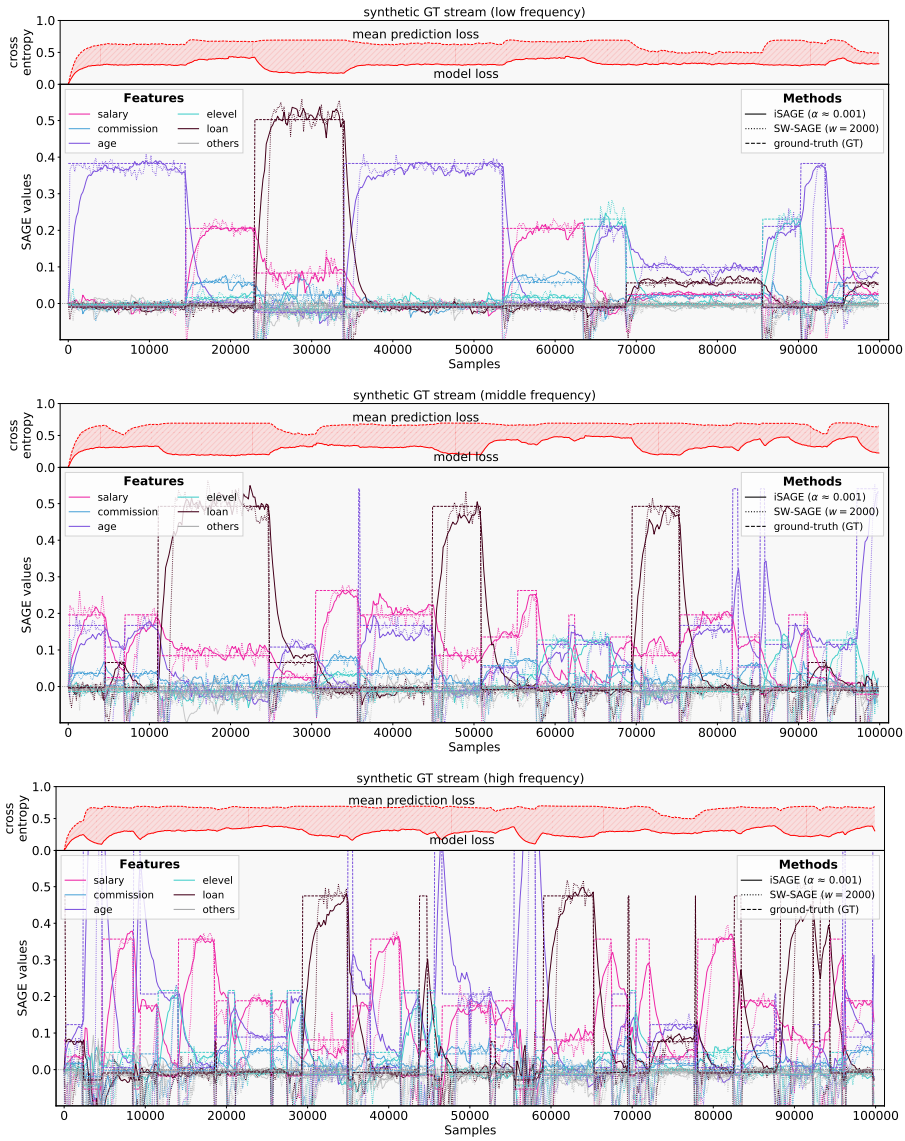

Fig. 9. Three exemplary synthetic GT streams with iSAGE (solid), SW-SAGE (dotted), and the GT (dashed) for a low frequency (top), middle frequency (middle), and a high frequency (bottom) scenario. Presented are coupled iSAGE and SW-SAGE explanations with $alpha \approx 0.001$, and $w = 2000$ and $m = 4$, respectively.

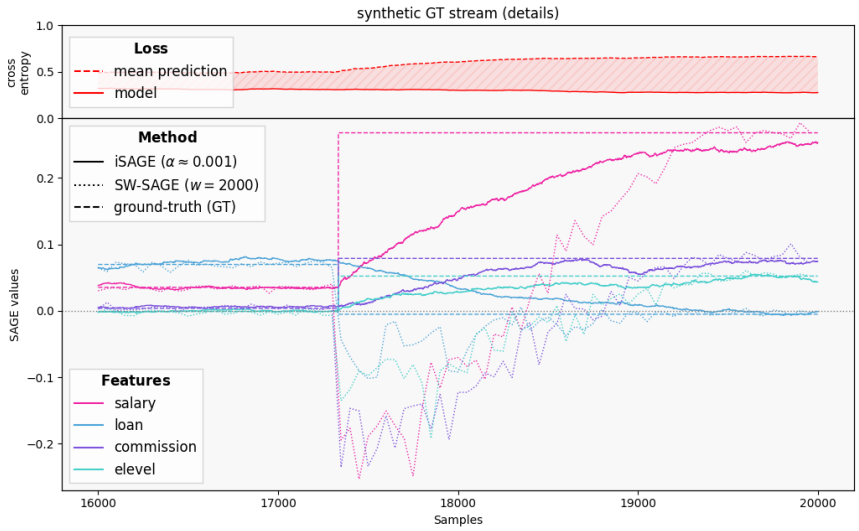

Fig. 10. Detail view of a synthetic GT data stream. The models switch after 17 335 samples (different GT values). Before the switch, iSAGE and SW-SAGE approximate the GT well. Yet, after the switch, SW-SAGE recovers more slowly with a high approximation error.

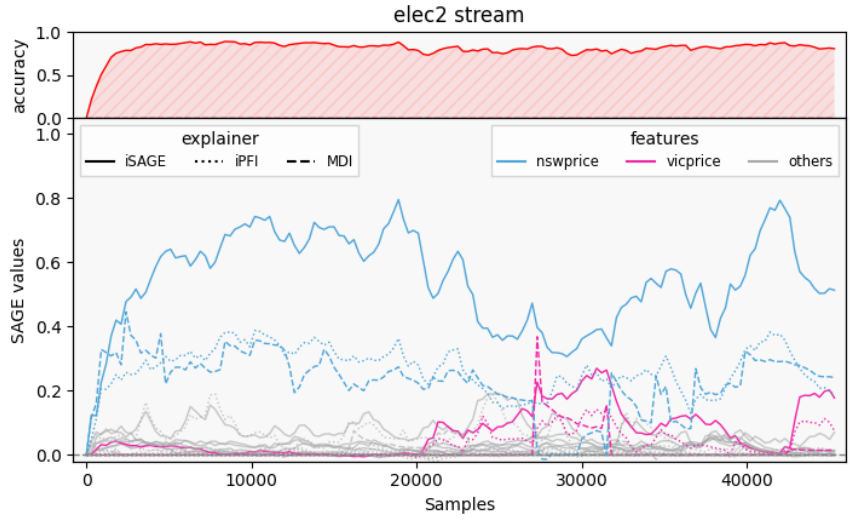

Fig. 11. iSAGE, iPFI, and MDI for an HAT on *elec2*

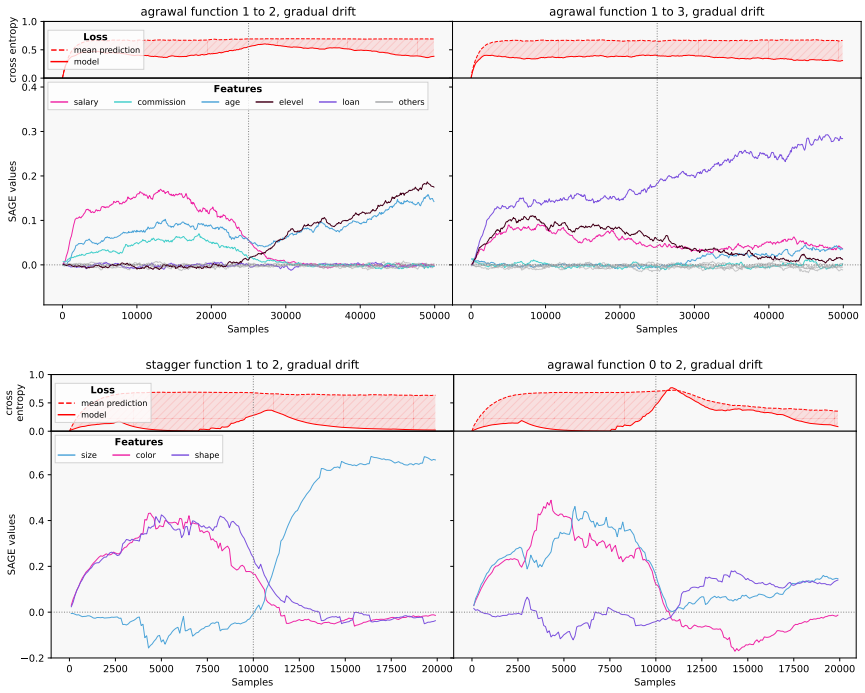

Fig. 12. Two example cases of iSAGE explaining an ARF on an *agrawal* gradual concept drift stream (top) and a HAT on a *stagger* gardual concept drift stream (bottom)

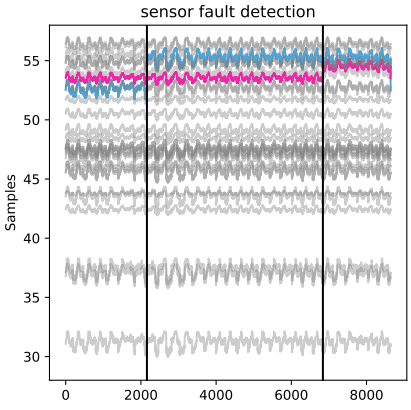

Fig. 13. Sensor readings of the simulated online sensor network.

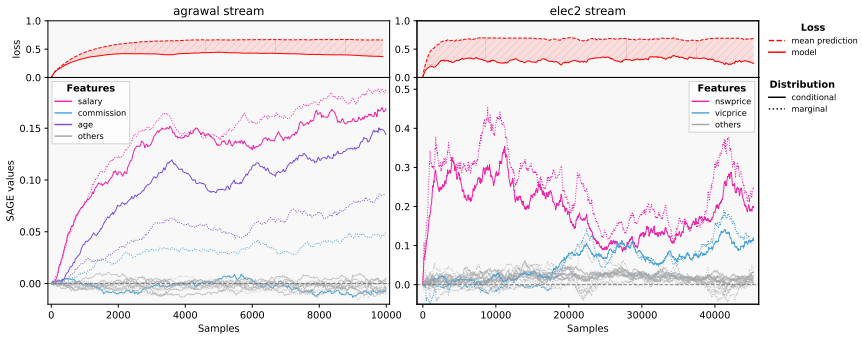

Fig. 14. Comparison of iSAGE with conditional (dotted) and marginal (solid) for an *agrawal* stream with known feature dependencies (left) and *elec2* (right).