ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

Real vs. Virtual Drift: Creating Realistic Stream Learning Benchmarks

Fabian Hinder, Johannes Brinkrolf, Kathrin Lammers, and Barbara Hammer *

Bielefeld University – Faculty of Technology
Inspiration 1, 33619 Bielefeld – Germany

Abstract. Concept drift – changes in the data distribution over time – is a central challenge in stream learning. However, existing benchmarks either lack controlled drift or fail to capture the characteristics of real-world data. We propose a pipeline for constructing verifiable and realistic drift, enabling more systematic evaluation of stream learning algorithms. Here, we pay special attention to controlling both real and virtual drift. To underscore the relevance of our contribution, we analyze the effects of real and virtual drift on both real-world and synthetic data streams using our method, revealing a substantial mismatch between the two setups.

1 Introduction

Data from the real world is often subject to ongoing changes known as concept drift [1, 2, 3], or drift for short. Drift may result from seasonal changes, changed demands, aging of sensors, etc. This is not well aligned with the classical batch setting of machine learning, where data points are assumed to be identically distributed between train and test set, creating challenges for maintaining high model performance [2]. In stream learning, this is approached by continuously updating models, ensuring their constant adaptation to the changing demands.

This introduces additional algorithmic components: memory buffer, update scheduling, drift detectors, etc. [2, 3]. Ideally, these components, their behavior, and effects should be subjected to systematic evaluation. However, while numerous synthetic generators and several real-world stream benchmarks exist [2, 3, 4], neither provides a satisfactory basis for such in-depth analysis. Synthetic generators tend to be overly simplistic and potentially unrealistic, whereas real-world streams lack controlled and verifiable properties.

In this work, we propose a systematic approach to create controlled and realistic drift scenarios based on real-world data streams. Our focus is on (i) *isolating* drifts in real-world data streams to serve as realistic sources (paying special attention to virtual and real drift), (ii) providing techniques to *verify* their quality and validity, and (iii) systematically *creating* streams with controlled drift based thereon. This allows the construction of realistic, controlled benchmarks and, by extension, allows studying drift and drift-related phenomena in the context of drift detection, stream learning, and model adaptation.

Based on those ideas, we compare the impact on performance for various models for real-world and synthetic data. We focus on real and virtual drift and show significant differences between real-world and synthetic streams. While real drift causes far more severe issues in synthetic data streams, which is in line

*Funding in the scope of the BMFTR project KI Akademie OWL under grant agreement No. 16IS24057A and the ERC Synergy Grant "Water-Futures" No. 951424 is gratefully acknowledged.

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

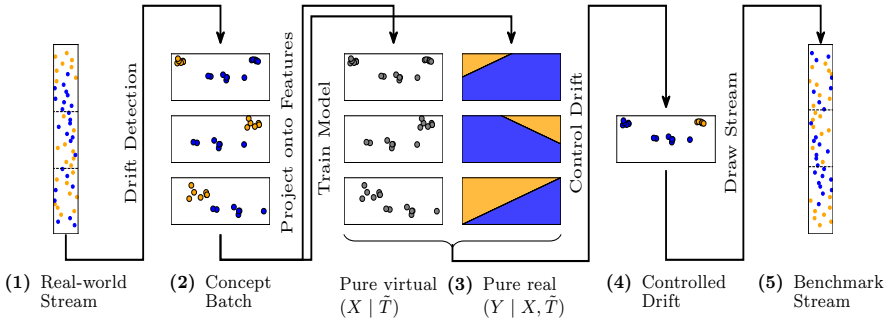


Fig. 1: Generating realistic controlled drift: 1) collect real-world stream, 2) isolate concepts from real-world stream, 3) separate real and virtual drift, 4) create concepts with fully controlled properties, 5) sample from concepts to create a realistic stream with controlled behavior

with the literature [2, 3], on real-world data streams, we see a far less severe, nearly opposite effect. This underscores the value of a systematic approach to ensure that efforts are directed toward actual core problems.

In the following, we first introduce the core concepts of drift and discuss how to control and verify it (Section 2). We then show the scientific relevance of our considerations by performing controlled analysis on synthetic and realistic data showing significant differences in model behavior (Section 3).

2 Creating Drift Benchmarks

In the following, we will discuss how to create a new, drifting data stream from an existing one in a step-by-step fashion. This includes algorithmic aspects on how to create the respective intermediate objects and final streams, as well as best-practice verification steps to ensure formal correctness of the stream. A schematic overview is presented in Fig. 1.

2.1 Isolating, Verifying, and Creating Drift

In this work, we mainly consider concept drift in the context of stream learning. A stream is an infinite sequence of independent observations $X_1, X_2, \ldots$, each drawn from a potentially different distribution $X_i \sim \mathcal{D}_i$ with drift referring to $\mathcal{D}_i \neq \mathcal{D}_j$ for some i, j [2]. Using a fully probabilistic framework [5], one augments each sample with a timestamp T_i, making drift equivalent to statistical dependence between data X and time T. As streams are assumed to be equidistantly sampled [2], by normalizing, we thus assign $T_i = i/N \in [0, 1]$, creating such a dated data stream [1, 5].

Isolation: Starting with a real-world data stream, we usually know little about its drift due to its statistical nature. Background knowledge can suggest potential drift points, but reconstructing exact samples is difficult and cannot rule out additional, unexpected drifts. Also, when drift occurs gradually, attribution to a stable concept is impossible.

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

Drift detectors can be used to find the most promising candidates. Unsupervised, offline, and multi-change point detectors are preferable, due to better statistical guarantees (see [1] "block-based detectors"). Those detectors are provided with label-data-point-pairs to also detect real drift [6].

Verification: A simple way to remove all potential drift in a batch is to randomly permute samples within it. While this eliminates remaining drift, starting from low-drift batches preserves realism. To verify that batches differ, pairwise two-sample tests like [7] are recommended. Those usually have higher statistical power. For many batches, a suitable α-level correction, such as Bonferroni, is necessary.

Creation: Given drift-free batches, those can be combined into data streams. Here, both the moment and the dynamic of the drift are controlled as well as the statistical properties. The most discussed statistical property of drift is whether it is virtual or real, which we will discuss in the next section. Drift dynamics [2, 3] describe how we transition from sampling from one batch to the next (abrupt, gradual, incremental, reoccurring). Except for incremental drift, all of those are obtained by sampling for the respective batches with the corresponding frequencies.

2.2 Controlling Real vs. Virtual Drift

One of the most considered aspects of drift is the subdivision into virtual – a change in the data distribution $\mathcal{D}_t(X)$ – and real drift – a change in the conditional label distribution $\mathcal{D}_t(Y \mid X)$ [3, 2]. An adaptation to the statistical framework of [5] was provided by [6], relating real drift to conditional dependence. Real drift is widely considered as the more consequential, while virtual drift is often treated as secondary [3, 2], due to the different effects both have on the decision boundary.

While considered central, naïve approaches to induce purely real or purely virtual drift usually fail. Techniques such as (sub-regional) sub-sampling (for virtual drift) or relabeling, as label flips (for real drift) to data streams without drift, can create pure but unrealistic drift. Similarly, label-aware sub-sampling typically induces virtual drift due to existing correlations. Furthermore, when using single concepts as before, these likely contain both real and virtual drift between them, making controlled drifting behavior hard to achieve.

Isolation: To address this problem, we propose learning the temporal-conditional distribution $\mathcal{D}_t(Y \mid X)$ in a batch setup fashion. After splitting the stream into batches, we obtain dated-data-label triples $(\tilde{T}_i, X_i, Y_i)$, where X_i denotes the data, Y_i the labels, and $\tilde{T}_i$ a batch identifier. According to [5, 6], these triples can be considered as i.i.d. samples from the joint distribution of labels, data, and time. Thus, general-purpose methods that perform well on a wide range of different data types, like random forests, extra tree forests, or XGBoost, offer good options to estimate $\hat{\mathbb{P}}_{Y|\tilde{T},X}$ in the usual batch fashion. The used model should be sufficiently powerful to learn the decision rule and to provide enough information to sample labels from it, i.e., in the case of classification, to yield accurate class probabilities rather than only label predictions.

Creation: By labeling the data in the batches using those models, we separate the virtual drift between the batches from the real drift encoded by the model. We thus turn n batches with unknown drifting behavior into n^2 batches

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

with known drifting behavior. Using those batches, we proceed as described in the last section.

Verification: By applying standard techniques such as cross-validation, we can evaluate the quality of the "simulation." We would like to point out that for realistic stream well-performing models are required. The existence of real drift is assessed by comparing model outputs for the respective data points across different batches, while virtual drift is tested using two-sample tests as before. This allows us to ensure sufficiently severe drift between batches.

3 Empirical Evaluation

In the following, we will study the effect of the different drift types on model performance. We focus on two aspects related to real and virtual drift: 1) how well do synthetic data streams resemble real-world data streams in terms of model performance, and 2) does the claim that real drift is more consequential for model performance compared to virtual drift hold up. For simplicity, we will not consider a direct streaming setup, but a simpler train-test setup, which is important for stream learning but also for transfer learning, where similar questions are considered.[1]

Data streams and learning models We use the standard synthetic data-streams Agrawal, Mixed, RandomTree, SEA, and Sine [4]. As these only show real drift, we create additional virtual drift by regional sub-sampling where the decomposition into regions corresponds to Voronoi cells induced by a k-means.

For real-world data streams, we use Credit Card, Electricity, Http [4], Forest Cover Type [8], and Nebraska Weather [9]. Here, we only use the stream's native virtual drift. For the real drift, we use the natural drift (learned by a random forest) and optional label flips to enhance the drift (again based on k-means). In cases of Http and Credit Card, we only use the label flip as the natural real drift is too weak.

Throughout our experiments, we use the following models: logistic regression (LR), multi-layer perceptron (1 hidden layer, 100 neurons; MLP), random forest (RF), linear SVM (SVM), k-nearest neighbor (k-NN) [10].

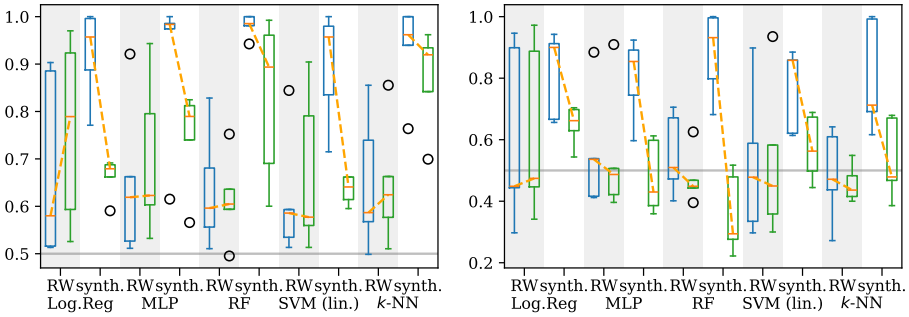
Impact of Drift on Performance To evaluate the effect of drift on model performance, we use the classical train-test setup. We train the models (400 data points) and then test them on three batches (no, real, virtual drift; 200 data points each). We use the Matthews Correlation Coefficient (MCC) to compensate for class imbalance and repeat the procedure 500 times per stream. We also considered balanced accuracy and F1-scores leading to similar results which we do not present due to space restrictions.

Since we use the same models for all three test sets, we can directly compare the performance. Fig. 2a shows the probability of a performance decrease per model and drift type compared to the non-drifting test set. Here, each box plot represents all data sources of the respective category (real-world/synthetic).

As can be seen, all types of drift lead to a significant decline in performance in all models and across all data sources. However, while for synthetic data sources the effect is very strong and stronger for real drift, we see much weaker effects on real-world-based data. Furthermore, virtual drift tends to have a (slightly)

[1] Experimental code `https://github.com/FabianHinder/Creating-Drift-Benchmarks`

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.



(a) Decrease induced by drift between train and test dataset.

(b) Decrease induced by impurity in training set; no additional drift.

Fig. 2: Probability of performance changes; each point represents one dataset (500 repetitions). Grouped by model, data stream type (background color: gray for real-world (RW); white for synthetic), and drift type (left/blue for real; right/green for virtual). Dashed orange lines indicate median change comparing real and virtual drift setup.

stronger effect compared to the real drift for all models but SVMs. Interestingly, this effect persists even though the real drift was amplified while the virtual drift remains in the natural state. If we only consider the three non-flip datasets, the effect is clearly visible for all models. This can be explained by the relative simplicity of the synthetic data streams regarding the complexity of the decision function, which is supported by the fact that the absolute scores are significantly higher for the synthetic compared to the pure and flipped data streams. Overall, this shows limitations of current synthetic data streams for studying algorithms.

Effect of Impure Training Samples We further investigate the effect of mixed training samples. This can occur when drift detectors select inappropriate split points and do not discard enough outdated data. We make use of a similar setup, but now compare the performance of the model trained and tested on the same dataset and compare against a model trained on an extended train set, which is obtained by adding 200 data points of a drifted distribution (test and train remain independent; total train size 600 data points). Importantly, the models have more information available which, however, is also subject to more variation and potential inconsistencies.

The results are shown in Fig. 2b. As before, we see a clear difference between the profiles of real-world and synthetic data sources. For synthetic data, in all cases but one, more than half of all data streams have a chance of a decrease in performance of over 85% for real drift, and in all but two cases, the effect is below 50% (random chance) for virtual drift. In the case of random forests, the effect is nearly reversed. Furthermore, the profile for virtual drift is not evenly distributed, while the median for more complex models (MLP, RF, k-NN) is always below 50%, for the simpler models (LR, SVM) it remains above 50%. This can be explained by the limited model complexity. This finding shows that general statements about the type of drift cannot be made, but considering the model class is paramount.

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

For real-world data, the difference in effect size between the drift types is significantly smaller. Only considering the non-flipped real-world data sets, all data streams remain below 50%, i.e., we see an increase in performance despite drift. This counterintuitive observation can be explained by a small difference between theoretically optimal models and a reduction in training error, resulting in an overall better outcome. The discrepancy induced by the drift type is less severe compared to the synthetic data streams. While there is a trend, it is mainly carried by the flip examples. In fact, for SVM and k-NN, we see the opposite effect, which underscores that, in particular for analytical studies, considering more realistic data streams is essential.

4 Conclusion

In this paper, we presented a method to create more realistic drift benchmarks for stream learning, focusing in particular on creating real and virtual drift in a controlled manner. We discussed options to verify the created benchmarks. Based thereon, we showed that there is a significant difference between real-world-based and common synthetic data streams regarding the effects of drift on performance. Our findings are highly relevant as they suggest that the folklore that virtual drift is secondary to real drift in relevance does not hold up. Furthermore, we observed that, for universal models, the occurrence of drift in the data does not strictly necessitate retraining, but that even drifted samples can still be useful for model training, questioning another widely believed statement. As both statements together have been instrumental for a lot of research, this shows that controllable and realistic data streams are of paramount importance to facilitate a full understanding of the properties of stream learning algorithms and ensure the improvement of the field.

References

[1] F. Hinder, V. Vaquet, and B. Hammer. One or two things we know about concept drift—a survey on monitoring in evolving environments. part a: detecting concept drift. *Frontiers in Artificial Intelligence*, 7:1330257, 2024.

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

[3] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang. Learning under concept drift: A review. *IEEE transactions on knowledge and data engineering*, 31(12):2346–2363, 2018.

[4] J. Montiel, M. Halford, S. M. Mastelini, G. Bolmier, R. Sourty, R. Vaysse, A. Zouitine, H. M. Gomes, J. Read, T. Abdessalem, and A. Bifet. River: machine learning for streaming data in python. *Journal of Machine Learning Research*, 22(110):1–8, 2021.

[5] F. Hinder, A. Artelt, and B. Hammer. Towards non-parametric drift detection via dynamic adapting window independence drift detection (dawidd). In *International Conference on Machine Learning*, pages 4249–4259. PMLR, 2020.

[6] F. Hinder, V. Vaquet, J. Brinkrolf, and B. Hammer. On the hardness and necessity of supervised concept drift detection. In *ICPRAM*, pages 164–175, 2023.

[7] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *The journal of machine learning research*, 13(1):723–773, 2012.

[8] Jock A. Blackard, Denis J. Dean, and Charles W. Anderson. Covertype data set, 1998.

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

[10] F. Pedregosa et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.