Tracking Change on the Edge

Enrique Nueve
NAISE
enrique.iv@northwestern.edu

Nicola J. Ferrier
Argonne National Laboratory
nferrier@anl.gov

Sean Shahkarami
Argonne National Laboratory
sshahkarami@anl.gov

Seongha Park
Argonne National Laboratory
seongha.park@anl.gov

Abstract

In deployment, it is crucial to monitor the effects of Concept Drift on a model. However, for Deep Learning models that may use an array of higher-dimensional data, the traditional Statistical driven method for Concept Drift become infeasible given they were constructed for use on low dimensional data. In this work, we propose a methodology using Normalizing Flows to transform complex distributions of feature maps to simple prior distributions, allowing for the use of traditional Concept Drift detection methodology. Our proposed method has the benefits of being able to be used on a large array of Deep Learning models and easy interpretation of whether damaging Concept Drift is occurring.

1 To do

1. I have no clue

2 Journal

Upon discussing with Sandeep, going about in a confidence estimation or out of sample detection may be the way to go. In hand, the reading in concept drift and normalizing flows may not be of use. The work in density estimation may be of use, I could add that into the testing section of out of samples.

1. Detect samples out of distribution or of low confidence
2. Experiment design, which samples to send back to cloud
3. Response to predictions sent back to cloud

3 Open Questions

1. What confidence measures will I use?
2. What samples do I send back to the edge
3. What data sets will I use for test
4. What metrics can I use for test?

4 Implemented methods

1. Normalizing Flows: NICE, Real NVP, MADE

Preprint. Under review.

2. Concept Drift: Concept drift test from Reactive Soft Prototype Computing for Concept Drift Streams
3. IN PROGRESS: Hint, Glow, Real NVP for images

5 Overview

5.1 Types of Concept Drift

1. **Type 1 (Virtual)**: $P_t(X) \neq P_{t+1}(X)$ yet $P_t(Y|X) = P_{t+1}(Y|X)$
2. **Type 2 (Real)**: $P_t(X) = P_{t+1}(X)$ yet $P_t(Y|X) \neq P_{t+1}(Y|X)$
3. **Type 3 (Real)**: $P_t(X) \neq P_{t+1}(X)$ and $P_t(Y|X) \neq P_{t+1}(Y|X)$

6 Related work

6.1 Theory Concept Drift

1. **An Information-Theoretic Approach to Detecting Changes in Multi-Dimensional Data Streams**
   - **Summary**: Uses ideas from information theory to provide learning guarantees
2. **ENSURING LEARNING GUARANTEES ON CONCEPT DRIFT DETECTION WITH STATISTICAL LEARNING THEORY**
   - **Summary**: Uses ideas from Statistical Learning Theory and Chaos Theory to provide a framework for concept drift detection to follow PAC learning.
3. **On learning guarantees to unsupervised concept drift detection on data streams**
   - **Summary**: Uses ideas from algorithmic stability to provide learning guarantees for unsupervised drift detection.

6.2 Confidence and Out of distribution detection

1. **Can You Trust Your Model's Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift**
   - NAISE_2021/EdgeMonitor/reviewed_papers/DataShift_Papers
   - **Summary**: Compares methods on how confidence changes on data from OOD (out-of-distribution)
   - **Probing questions**: It assumes that there is a relationship between confidence and accuracy
   - **Limits**: Shows that most methods don't work well on OOD
   - **Potential Improvements**:
2. **A statistical theory of out-of-distribution detection**
   - NAISE_2021/EdgeMonitor/reviewed_papers/DataShift_Papers
   - **Summary**:
   - **Probing questions**:
   - **Limits**:
   - **Potential Improvements**:
3. **Energy-based Out-of-distribution Detection**
   - **Summary**:
   - **Probing questions**:
   - **Limits**:
   - **Potential Improvements**:

6.3 Normalizing Flows

1. **NICE: Non-linear Independent Components Estimation** [DKB15]
   - **Summary**: Proposes a method using inspiration from the Method of Transformation of Random Variables to transform a complex data distribution to a simple prior distribution. The method is fully invertible, allowing for sampling of the prior to generate new samples.
   - **Probing questions**: What is the nature of the outliers when outputted from the prior? Do most samples stay near the mean?
   - **Limits**: The transformed dimension of the data must be the same as the original. The data must be shaped as a vector.
   - **Potential Improvements**: NICE is sensitive to training configuration. A decaying learning rate helped solve this issue. To generate samples, the variance of the prior should also be decreased in order to get clear data.
2. **Density estimation using Real NVP** [DSB17]

- **Summary**: Uses two neural networks in couple layers. Also proposes a new way to do batch norm to improve stability of training. Proposes different way to apply mask: spatial check board and halves. Real NVP uses a affine (scale and translate) coupling layer while NICE uses an additive coupling layer.
  Paper is not that good at explaining the archtiecture. There is sample code in keras here https://keras.io/examples/generative/real_nvp/ .
- **Probing questions**: Could normalizing flows be used for time-series forecasting?
- **Limits**: Seems like shape of network would have to be re-fitted for different datasets pretty extensively.
- **Potential Improvements**: Try to be able to change output dim size compared to input

3. **Video on Normalizing Flows https://www.youtube.com/watch?v=u3vVyFVU_lI**
   - **1**: Talks about paper called HINT which allows for a fuller triangular jacobian determinant matrix. Uses an iterative scheme of coupling layers to do this.
   - **2**: Autoregressive model: inspired by product rule of probability. Can combine to form Autoregressive Models as Flows. **READ, Masked Autoregressive Flow for Density Estimation**, https://arxiv.org/abs/1705.07057
   - **3**: Multi-scale flows, drop operations on dimensions to lessen computation. This is shown in Real NVP with the spatial checkboard and squeeze operations.
   - **4**: Continuous-time Normalizing Flows is talked about in paper **FFJORD**. Uses solution of ODE as a flow.

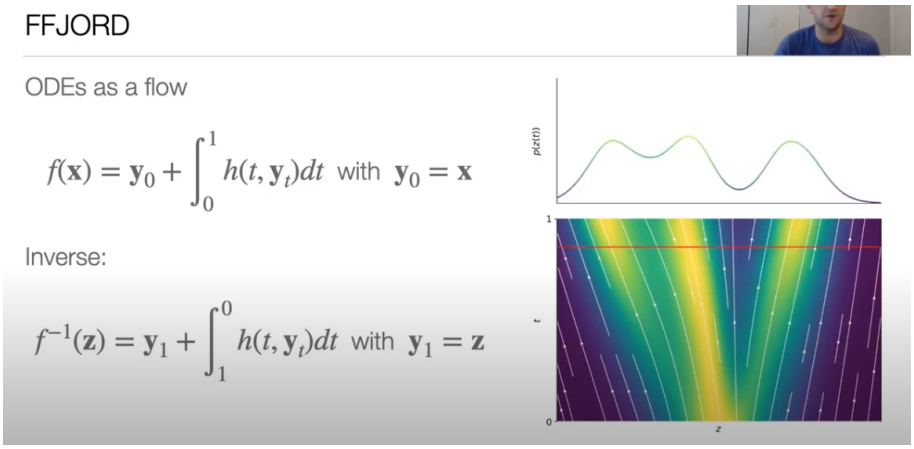
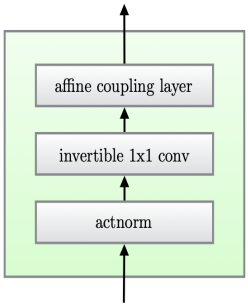


Figure 1: As t increases the prob dist changes towards prior!

- **Glow: Generative Flow with Invertible 1x1 Convolutions** [KD18]
  - **Summary**:
    - Each step of flow consists of actnorm (Section 3.1) followed by an invertible 1 × 1 convolution (Section 3.2), followed by a coupling layer (Section 3.3).
- **Uncertainty quantification in medical image segmentation with normalizing flows**
  - **Summary**: We propose a novel conditional normalizing flow model – cFlow Net – and demonstrate the use of two types of normalizing flow trans- formations: Planar flows [19] and Generative Flows
    *
  - **Probing questions**:
  - **Limits**:
  - **Potential Improvements**:
- **LEARNING LIKELIHOODS WITH CONDITIONAL NORMALIZING FLOWS**
  - **Summary**:
    *



(a) One step of our flow.

Figure 2: Glow couple diagram

  - **Probing questions**:
  - **Limits**:
  - **Potential Improvements**:

6.4 Concept Drift Detection

1. **Reactive Soft Prototype Computing for Concept Drift Streams** [RHS20]
   - **Summary**:
     - Drifts appear differently through speed, intensity, and frequency, i.e. incremental, abrupt, gradual or reoccurring.
     - Concept drift detectors are trying to detect a change in streams either by monitoring the distribution of the streams or the performance of a classifier with respect to some benchmark, e.g. accuracy.
     - Concept Drift: $\exists X : p_{t-1}(X, y) \neq p_t(X, y)$
     - uses a memory strategy to keep track off recent points from the stream. Uses two windows, one for recent points, and one for uniform samples over all stream.
     - Eq(5) in the paper gives the formula for the KS test for the two windows on one dim data. For higher dim data, test applied to all data dim.
     - By applying test to many dimensions, many false positives occur. This is addressed by using the **Bonferroni-Dunn correction**
   - **Probing questions**: How does the kolmogorov-smirnov (KS) test work on higher dimensional data?
   - **Limits**: How will I choose what to store in the sliding window given the dynamic nature of my images. The number of sliding windows must scale according to number of dimensions
   - **Potential Improvements**:
2. **Pocket Data Mining** [Gab13]
   - Notable success of the use of Hoeffding bound to approximate the data mining models for streaming data has been recognized.
   - tightly-coupled multiprocessor architectures all processors use the same shared memory, hence distributing data is not required.
   - loosely-coupled multiprocessor architecture each processor uses its own private memory.
   - The distributed analysis of local data sources allows to execute data mining algorithms concurrently, to combine their results and thus is speeding up the data mining process compared with the centralized approach.

- Mobile Agent Resource Discoverers (MRD) are mobile agents that are used to roam the network in order to discover for the data mining task relevant data sources, sensors, AMs and mobile devices that fulfill the computational requirements.

3. **Recent trends in streaming data analysis, concept drift and analysis of dynamic data sets**[BHS19]
   - Streaming data: data where measurements arrive continuously as a data stream.
   - stability-plasticity dilemma arises and learning faces an essentially illposed problem [12]: when is observed change caused by an underlying structure and should be taken into account, and when is it given by noise and should be neglected?
   - For supervised learning, one distinguishes the notion of real drift, which refers to a change of the posterior distribution P(y|x) and virtual drift or covariate shift, which refers to a change of the input distribution P(x) only without affecting the posterior.
4. **On the Reliable Detection of Concept Drift from Streaming Unlabeled Data** [SK17]
   - **Summary**:
     - Extending the notion of classifier uncertainty, as a uni-variate signal, for detecting concept drift.
     - Developing the Margin Density Drift Detection (MD3) algorithm, as an incremental streaming algorithm, for detecting drifts from unlabeled data.
     - Formulation of margin density for classifiers with explicit margins (such as SVM) and for those with- out explicit margins (such as Decision Trees), with comparison and empirical evaluation demonstrating their equivalence.
     - A novel drift induction framework for introducing concept drift into datasets, for controlled experimentation on a variety of data domains. The drift induction process provides for a reusable testing framework, by enabling the addition of concept drift to static datasets, for better experimentation and analysis of drift detection methodologies.
     - Experimental evaluation of the MD3 framework, on datasets from cybersecurity domains, highlighting the efficacy of the proposed approach in providing reliable and robust adversarial drift detection.u
     - **OTHER TEST FOR CONCEPT DRIFT** Kolmogorov-Smirnov test, Wilcoxon rank sum test and the two sample t-test, was suggested in (Dries and Ruckert, 2009)
     -

Implicit drift detection (Unsupervised)	Novelty detection/ clustering methods	OLINDDA (Spinosa et al., 2007), MINAS (Faria et al., 2013), Woo (Ryu et al., 2012), DETECTNOD (Hayat and Hashemi, 2010), ECSMiner (Masud et al., 2011), GC3 (Sethi et al., 2016b)
	Multivariate distribution monitoring	CoC (Lee and Magoules, 2012), HDDDM (Ditzler and Polikar, 2011), PCA-detect (Kuncheva and Faithfull, 2014; Qahtan et al., 2015)
	Model dependent monitoring	A-distance (Dredze et al., 2010), CDBD (Lindstrom et al., 2013), Margin (Dries and Rückert, 2009)

Figure 3: Unsupervised Concept Drift Methods

   - **Probing questions**: The Change of Concept(CoC) technique (Lee and Magoules, 2012) considers each feature as an independent stream of data and monitors correlation between the current chunk and the reference training chunk. Can this method be used with normalizing flow to make assumption valid?
   - **Limits**: Only works on models such as SVM and Decision Trees
5. **Hellinger Distance Based Drift Detection for Nonstationary Environments**[DP11]
   - **Summary**: Hellinger distance drift detection method (HDDDM) is a feature based drift detection method, using the Hellinger distance between current data distribution and a reference distribution that is updated as new data are received. The Hellinger distance is an example of divergence measure, similar to the Kullback-Leibler (KL) divergence

  - Hellinger distance is a bounded dis- tance measure: for two distributions with probability mass functions (or histograms representing these distributions) P and Q the Hellinger distance is $\delta_H(P,Q) \in [0, \sqrt{2}]$. If $\delta_H(P,Q) = 0$, the two probability mass functions are completely overlapping and hence identical. If $\delta_H(P,Q) = \sqrt{2}$, the two probability mass functions are completely divergent (i.e. there is no overlap).
- **Probing questions**: How does this perform on neural networks?

6. **Request-and-Reverify: Hierarchical Hypothesis Testing for Concept Drift Detection with Expensive Labels** [YWP18]
   - **Summary**: Semi-supervised concept drift method.
     - Two methods, namely Hierarchical Hypothesis Testing with Classifica- tion Uncertainty (HHT-CU) and Hierarchical Hy- pothesis Testing with Attribute-wise "Goodness-of- fit" (HHT-AG)
     -) a change in the marginal probability Pt(X); 2) a change in the posterior probability Pt(y|X). Existing studies in this field primarily concentrate on detecting posterior distribution change Pt(y|X), also known as the real drift [Widmer and Kubat, 1993]
     - The virtual drift detection, though making no use of true label yt, has the issue of wrong interpretation (i.e., interpret- ing a virtual drift as the real drift). Such wrong interpretation could provide wrong decision about classifier update which still require labeled data [Krawczyk et al., 2017].
     - The first method incrementally tracks the distribution change with the defined classification uncertainty measurement in Layer-I, and uses permutation test in Layer-II, whereas the second method uses the standard Kolmogorov-Smirnov (KS) test in Layer-I and two-dimensional (2D) KS test [Peacock, 1983] in Layer-II.
     - The problem of concept drift detection is identifying whether or not the source P (i.e., the joint distribution Pt(X,y)1) that generates samples in SA is the same as that in SB (even without access to the true labels yt) [Ditzler et al., 2015; Krawczyk et al., 2017].
     - **Methods for Concept Drift detection: Conjunctive Normal Form (CNF) density estimation test [Dries and Ru ckert, 2009] and the Hellinger distance based density estimation test [Ditzler and Polikar, 2011].**
     - The Confidence Distribution Batch Detection (CDBD) approach [Lindstrom et al., 2011] uses Kullback-Leibler (KL) divergence to compare the classifier output values from two batches.
     -
   - **Probing questions**: How could a user be alerted for potential concept drift? Could we send an email with a sample of images and predictions?
   - **Limits**: Layer I is automated but layer II requires labeling

7. **Learning under Concept Drift: A Review** This paper reviews over 130 high quality publications in concept drift related research areas, analyzes up-to-date developments in methodologies and techniques, and establishes a framework of learning under concept drift including three main components: concept drift detection, concept drift understanding, and concept drift adaptation. This paper lists and discusses 10 popular synthetic datasets and 14 publicly available benchmark datasets used for evaluating the performance of learning algorithms aiming at handling concept drift. [Lu+18]
   - **Summary**:
     -
   - Hierarchical drift detection is an emerging drift detection category that has a multiple verification schema. The algorithms in this category usually detect drift using an existing method, called the detection layer, and then apply an extra hypothesis test, called the validation layer, to obtain a second validation of the detected drift in a hierarchical way.
   - **Probing questions**:
     - How can my method inform what kind of drift is occurring?
     - How does my method say if drift is occurring?

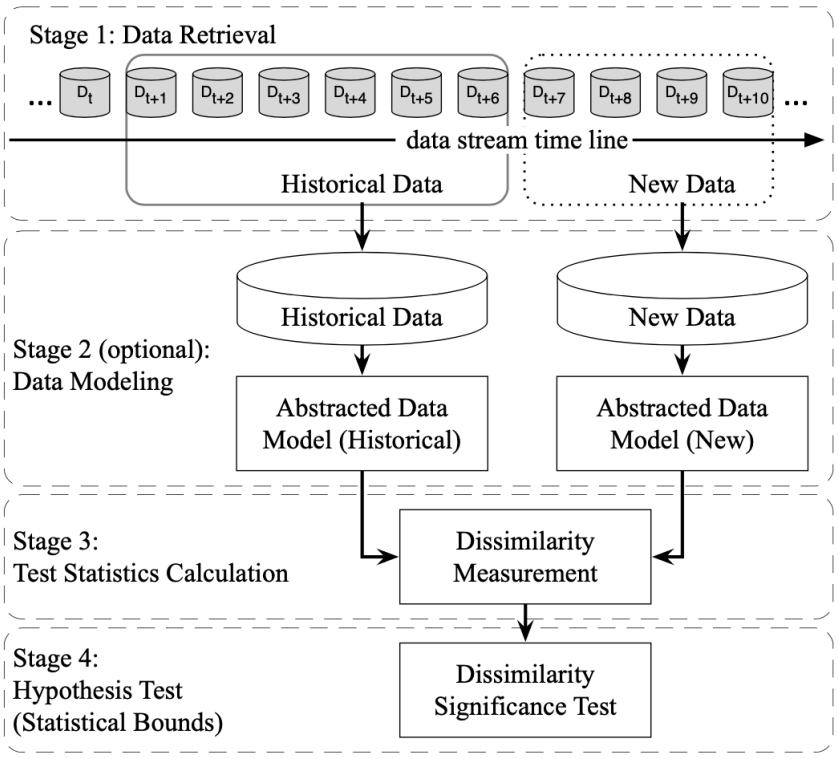


Figure 4: Concept drift diagram

  - *My normalize flow method could say if shift but not tell if difference between real or virtual drift, could human in the loop then decide if real drift or not?*
  - Is there Hierarchical drift literature that could be used to get bounds on my Human in the Loop idea
  - **holdout** should follow the rule: when testing a learning algorithm at time t, the holdout set represents exactly the same concept at that time t. Unfortunately, it is only applied on synthetic datasets with predefined concept drift times.
  - **Prequential** is a popular evaluation scheme used in streaming data. Each data instance is first used to test the learning algorithm, and then to train the learning algorithm. This scheme has the advantage that there is no need to know the drift time of concepts, and it makes maximum use of the available data. The prequential error is computed based on an accumulated sum of a loss function between the prediction and observed label: S = $\sum_{t=1}^{n} f(\hat{y}_t, y_t)$. There are three prequential error rate estimates: a landmark window (interleaved-test-then- train), a sliding window, and a forgetting mechanism [88].
  - **Controlled permutation** [89] runs multiple test datasets in which the data order has been permutated in a controlled way to preserve the local distribution, which means that data instances that were originally close to one another in time need to remain close after a permutation. Controlled permutation reduces the risk that their prequential evalua- tion may produce biased results for the fixed order of data in a sequence.

- **Evaluation metrics**:
  - RAM-hours
  - Kappa statistic
  - Kappa temporal statistic
  - Combined Kappa statistic
  - Prequential AUC

- true detection rate, 2) false detection rate, 3) miss detection rate, and 4) delay of detection [22].
- **Datasets**
  - Electricity: https://www.openml.org/d/151
  - Covertype: https://archive.ics.uci.edu/ml/datasets/covertype
  - STAGGER
  - LED
  - rotating chessboard
  - Sine
  - Waveform

8. **An overview of unsupervised drift detection methods**[Gem+20]
   - **Summary**:
     -
   - **Probing questions**:
   - **Limits**:
   - **Potential Improvements**:
9. **Accumulating regional density dissimilarity for concept drift detection in data streams**
   - **Summary**:
     - It consists of three components. The first is a k-nearest neighbor-based space-partitioning schema (NNPS), which transforms unmeasurable discrete data instances into a set of shared subspaces for density estimation. The second is a distance function that accumulates the density discrepancies in these subspaces and quantifies the overall differences. The third component is a tailored statistical significance test by which the confidence interval of a concept drift can be accurately determined.
     - According to the literature, a typical distribution-based detection method consists of three components. The first component is a data representation model through which critical information is retrieved and irrelevant details are discarded. The second component is a specific dissimilarity function designed to measure the discrepancies between the data models. One of the most natural notions for the distance between distributions is the total variation, or the L1 norm [33]. The third component is a statistical significance test. Statistical significance, namely the p-value, is the probability of obtaining the least extreme result given that a null hypothesis is true. In drift detection, the null hypothesis is true when the detected discrepancies are not caused by concept drift.
   - **Probing questions**:
   - **Limits**:
   - **Potential Improvements**:

7 Metrics for Concept Drift

1.

8 Methodology

1.

9 Experiments

After collecting the predictions of the primary model on both source and target datasets, we need to perform a statistical test to check if there is significant difference between the two distributions of predictions. One possibility is to use the Kolmogorov-Smirnov test and compute again the p-value, i.e., the probability of having at least such distance between the two distributions of predictions in case of absent drift. For this technique we are looking at the predictions (which are vectors of dimension K, the number of classes), and perform K independent univariate Kolmogorov-Smirnov tests. Then we apply the Bonferroni correction, taking the minimum p-value from all the K tests and requiring this p-value to be less than a desired significance level divided by K. Again we can assume there is a drift when the minimum p-value is less than the scaled desired significance level.

In order to benchmark the drift detectors in a controlled environment, we synthetically apply different types of shift to the electricity dataset:

- **Prior shift**: Change the fraction of samples belonging to a class (technically this could also change the feature distribution, thus it is not a 'pure' prior shift).
- **Covariate Resampling shift**: Could be either under-sampling or over-sampling. The former case consists in a different selection of samples according to the features values, for instance keeping 20 % of observations measured in the morning. The latter consists in adding artificial samples by interpolation of existing observations.
- **Covariate Gaussian noise shift**: Add Gaussian noise to some features of a fraction of samples.
- **Covariate Adversarial shift:** Amore subtle kind of noise, slightly changing the features but inducing the primary model to switch its predicted class. This type of drift is less likely to occur and difficult to detect, but its negative impact is large.

An ideal drift detector is not only capable of detecting when drift is occurring but is able to do so with only a small amount of new observations. This is essential for the drift alert to be triggered as soon as possible. We want to evaluate both accuracy and efficiency. We apply 28 different types of shifts to the electricity target dataset from the four categories highlighted in the previous section. For each shift situation, we perform 5 runs of drift detection by both domain classifier and black-box shift detector. For each run, we perform the detection comparing subsampled versions of the source and drifted datasets at six different sizes: 10, 100, 500, 1000, 5000, and 10000 observations.

10 Results

11 Conclusion

References

[DP11] Gregory Ditzler and Robi Polikar. "Hellinger distance based drift detection for non-stationary environments". In: *2011 IEEE symposium on computational intelligence in dynamic and uncertain environments (CIDUE)*. IEEE. 2011, pp. 41–48.

[Gab13] Mohamed Medhat Gaber. *Pocket data mining: techniques and applications*. 1st edition. Studies in big data 2. New York: Springer, 2013. ISBN: 9783319027104.

[DKB15] Laurent Dinh, David Krueger, and Yoshua Bengio. *NICE: Non-linear Independent Components Estimation*. 2015. arXiv: 1410.8516 [cs.LG].

[DSB17] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. *Density estimation using Real NVP*. 2017. arXiv: 1605.08803 [cs.LG].

[SK17] Tegjyot Singh Sethi and Mehmed Kantardzic. "On the reliable detection of concept drift from streaming unlabeled data". In: *Expert Systems with Applications* 82 (2017), pp. 77–99.

[KD18] Diederik P. Kingma and Prafulla Dhariwal. *Glow: Generative Flow with Invertible 1x1 Convolutions*. 2018. arXiv: 1807.03039 [stat.ML].

[Lu+18] Jie Lu et al. "Learning under concept drift: A review". In: *IEEE Transactions on Knowledge and Data Engineering* 31.12 (2018), pp. 2346–2363.

[YWP18] Shujian Yu, Xiaoyang Wang, and José C Prıncipe. "Request-and-reverify: Hierarchical hypothesis testing for concept drift detection with expensive labels". In: *arXiv preprint arXiv:1806.10131* (2018).

[BHS19] Albert Bifet, Barbara Hammer, and Frank-Michael Schleif. "Recent trends in streaming data analysis, concept drift and analysis of dynamic data sets." In: *ESANN*. 2019.

[Gem+20] Rosana Noronha Gemaque et al. "An overview of unsupervised drift detection methods". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 10.6 (2020), e1381.

[RHS20] Christoph Raab, Moritz Heusinger, and Frank-Michael Schleif. "Reactive Soft Prototype Computing for Concept Drift Streams". In: *Neurocomputing* 416 (Nov. 2020), pp. 340–351. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2019.11.111. URL: http://dx.doi.org/10.1016/j.neucom.2019.11.111.