2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)
October 11-14, 2020. Toronto, Canada

Naïve Approaches to Deal With Concept Drifts

Paulo R. Lisboa de Almeida
Department of Computer Science
Univ. do Estado de Santa Catarina
Joinville (SC), Brazil
paulo.almeida@udesc.br

Luiz S. Oliveira
Department of Informatics
Univ. Federal do Paraná
Curitiba (PR), Brazil
luiz.oliveira@ufpr.br

Alceu de Souza Britto Jr., Jean Paul Barddal
Graduate Program in Informatics (*PPGIa*)
Pontifícia Universidade Católica do Paraná
Curitiba (PR), Brazil
{alceu, jean.barddal}@ppgia.pucpr.br

***Abstract*—A common problem in machine learning is to find representative real-world labeled datasets to put the methods to test. When developing approaches to deal with concept drifts, some datasets such as the Forest Covertype and Nebraska Weather are common choices for testing, even though there is no consensus on whether these exhibit concept drifts or not. We argue that some well-known real-world concept drift datasets present a high serial dependence in the target class and may have only minor changes. With this in mind, we propose the use of naïve methods that should be used for comparison with methods that deal with concept drifts. The experimental results using six real-world well-known concept drift datasets show that the naïve approaches can be better than some methods to deal with possible concept drifts in datasets such as the Forest Covertype, Electricity, and Nebraska Weather. These results suggest that some widely used datasets may be trivial from the concept drift standpoint, and thus, should be avoided, or at least the results should be compared with the proposed naïve methods.**

***Index Terms*—concept drift, dataset, benchmark**

I. Introduction

Over the last years, many methods to deal with the Concept Drift problem have been proposed by the scientific community. Testing these methods impose a real and recurrent problem in machine learning: to find representative real-world datasets to stress such methods and analyze their behavior.

Many authors use artificial problems to test their methods, such as the STAGGER [1] and the Moving Hyperplane [2] datasets. These problems usually have low-dimensionality and represent some specific concept drifts, which may not be representative of real-world problems [3], [4]. Conversely, real-world scenarios often have high dimensionality, and the properties of the concept drifts, e.g., the type of change, rate of change, and the number of concept drifts, are unknown.

Real-world benchmarks used for testing methods that deal with concept drift include the Electricity, Forest Covertype, and Airlines datasets. Inspired by [4]–[6], we argue that some of the existing real-world benchmarks may present trivial problems due to temporal dependencies in the target classes. With this in mind, we present an analysis of the serial dependence, and the concept drift recovery difficulty of six real-world concept drift datasets commonly often employed.

We use Cramér's V test under several time lags to compute the serial dependence of the target class for each dataset. To obtain insights on the changes present in each dataset, we propose a set of naïve strategies, including (i) random triggers and (ii) static classifiers. The first are tailored to adapt to concept drifts, while the second are unable to handle concept drifts. Naïve methods should give us insight about the changes since these may not adapt to drifts, and thus, should not perform well, for instance, with relevant accuracy drops when concept drift occurs. The Cramér's V analysis also shows that all tested datasets have some kind of time dependence, and thus a blind classifier that labels an instance according to the label of its previous one should be used as a baseline when employing the test-then-train protocol.

We also use a random trigger, an online learner, and a classical static classifier as naïve approaches, and show that the accuracy rates of these methods can surpass some robust approaches that deal with concept drifts. This intriguing result suggests that the changes in the tested real-world datasets are minor since naïve methods recover from such changes. Therefore, new and more representative real-world concept drifts datasets are needed. This result also indicates that the naïve methods should be used in the tests when a new method to deal with concept drifts is proposed, in order to compare the proposed approach with naïve approaches and/or traditional machine learning methods.

The remainder of this paper is organized as follows. In Section II the concept drift problem and the Cramér's V test are defined. Section III presents related works that assess the quality and methodologies applied in the real-world datasets. Section IV describes the datasets used in this work. These datasets are analyzed in Section V, combined with the proposal of naïve methods to assess the quality of the methods and datasets. Section VI presents a proof of concept experiment considering two well-known methods that are compared with the proposed naïve approaches. Finally, our conclusions are presented in Section VII.

II. Definitions

A. Concept Drift

Consider $P_t(y)$ as the class *a priori* probabilities at a given time t, and $P_t(\mathbf{x})$ the unconditional distribution at the time t, a virtual concept drift may occur when between timestamps t and $t+\delta$, with $\delta \geq 1$, $P_t(y) \neq P_{t+\delta}(y)$ and/or $P_t(\mathbf{x}) \neq P_{t+\delta}(\mathbf{x})$ while the *a posteriori* probabilities $P(y|\mathbf{x})$ remain unaltered, that is, $P_t(y|\mathbf{x}) = P_{t+\delta}(y|\mathbf{x})$ [7]–[9].

On the other hand, a real concept drift is caused by a change in the *a posteriori* probabilities over time, thus $P_t(y|\mathbf{x}) \neq P_{t+\delta}(y|\mathbf{x})$, where the *a priori* probabilities and

the unconditional distributions may or may not change over time. Note that in a real concept drift scenario, the instances may change their target classes over time [7]–[9].

B. Cramér's V

A categorical time series with discrete time range can be seen a sequence $(X_t)_{\mathbb{N}} = \{x_1, \ldots, x_T\}$ of random variables, where the range $S = \{s_0, s_1, \ldots, s_d\}$ of X_t belongs to one of a finite number of unordered $d+1$ categories [10]. We consider the original series and its lagged version as a pair (X_t, X_{t-k}), of the time series as $(X_t, X_{t-k}) = (x_{k+1}, x_1), \ldots, (x_T, x_{T-k})$, where $k \in \mathbb{N}$. In other words, we shift the series in time by k [11].

Consider $\mathbf{Y}_t$ as a binary random vector, consisting of the vectors $\mathbf{e}_0, \ldots, \mathbf{e}_d \in \{0,1\}^{d+1}$, where $\mathbf{Y}_t = \mathbf{e}_j$ if $X_t = s_j$, and $\mathbf{Y}_t$ is the binary representation of X_t [11]. Given that $p_{i,j\in S}(k) = 1/(T-k)\sum_{t=k+1}^{T} Y_{t,i}Y_{t-k,j}$ is the lagged bivariate probability, and $\pi_i = (1/T)\sum_{t=1}^{T} Y_{t,i}$, the sample version of the Cramér's V for a lag k is given by

$$v(k) = \sqrt{\frac{1}{d}\sum_{i,j\in S}\frac{(p_{i,j}(k) - \pi_i\pi_j)^2}{\pi_i\pi_j}} \quad (1)$$

where $v(k) \in [0;1]$ indicates a perfect serial independence/dependence at lag k. A perfect serial dependence at lag k indicate that we can perfectly predict X_t after observing X_{t-k} [10].

III. Related Work

Besides the wide use of the real-world datasets in the literature, for the best of our knowledge, few works assess the quality and the methodology used in the tests when considering these benchmarks as concept drift scenarios. In the remainder of this paper, we refer to a classifier that predicts the label of an instance arriving at a timestamp t as the same observed at a timestamp $(t-1)$ as a blind classifier.

In [5] it is demonstrated that the Electricity and Forest covertype benchmarks present a high correlation in the target class attribute due to time dependence, and thus, a blind classifier yields better results than sophisticated methods developed to deal with concept drifts.

A similar statement was given in [12], where authors showed that a blind classifier achieves high accuracy in the Electricity benchmark when using the test-then-train approach and suggest that methods using the Electricity dataset should be compared against a blind classifier.

More recently, [6] showed that a blind trigger (that fires periodically) can achieve better results than some state-of-the-art triggers in the Electricity and Forest covertype benchmarks. The authors argue that this is caused by temporal dependencies in the datasets, and suggests that the final accuracy should not be the only metric to assess methods that deal with concept drifts. The authors suggest the use of artificial datasets and metrics such as the *Mean Time between False Alarms*, and the *Missed Detection Rate*, which measures the frequency of false alarms, and the rate of concept drifts not detected, respectively.

Finally, the authors in [13] developed metrics that are applied in time windows and can be used to take measurements such as the magnitude and duration of changes in $P(y)$, $P(\mathbf{x})$ and $P(y|\mathbf{x})$. The work of [14] extends the proposal of [13] to include more computation efficient metrics and analyzes some real-world benchmarks, such as Electricity and Airlines. Results show that the concept drifts present in the Electricity dataset are mostly virtual, and in the Airlines dataset, both virtual and real concept drifts are present.

IV. Concept Drift Benchmarks description

In this section, we briefly describe the real-world benchmarks commonly used for testing methods that deal with concept drifts[1]. We also include a description of the STAGGER and Moving Hyperplane artificial datasets, which are used in the analysis to compare the results.

STAGGER Concepts: This artificial dataset, proposed in [1], is composed of three attributes ($color, shape$ and $size$) and two classes ($positive/negative$). There are three concepts where the positive class is defined as given: in the first concept it is defined as $color = red \wedge size = small$; in the second it is given by $color = green \vee shape = circle$, and in the third concept the positive class is defined as $size = medium \vee size = large$. In this paper, we generate 6,000 samples, with a change of concept for every 2,000 samples.

Moving Hyperplane: Introduced in [2], this artificial dataset consists of a $d - dimensional$ real space containing samples generated uniformly in a predefined range. Samples are labeled as *positive* if $\sum_{i=1}^{d} w_i x_i > w_0$, otherwise the samples are labeled as *negative*. In the equation, w_i and x_i correspond to the i^{th} attribute weight and value, respectively. Concept drifts are introduced by varying the w_i weights according to $w_i = w_i + c\sigma$, where c controls the change, and σ controls the probability that the direction of change is reversed [15]. In this paper we consider $d = 5$, $c = 0.001\%$, $\sigma = 1\%$, and a total of 100,000 instances generated.

Electricity: Proposed in [16], this dataset refers to the Australian New South Wales Electricity Market. The dataset is composed of 45,312 samples, each of them containing five attributes. The task is to predict if the electricity price is higher or lower than a moving average of the last 24 hours.

Forest covertype: This dataset is composed of 581,012 samples and defines the classification task as identifying the forest covertype for $30m^2$ cells. Each instance is described by 54 attributes and a target class, which belongs to the range $[1, 7]$ that identifies the forest covertype [17].

Nebraska Weather: This dataset contains weather data collected by the U.S. National Oceanic and Atmospheric Administration in the Offutt Air Force Base in Bellevue, Nebraska. Each instance is composed of 8 attributes and a binary target class indicating the presence or not of rain. We tested the same configuration used in [18] and [19], where missing values were replaced by the mean of the features in the preceding and following samples. It contains 18,159 samples.

[1]Datasets available at http://prlalmeida.com.br/driftDatasets.html

Airlines: This dataset is a variation of the original dataset proposed in [20], where the task is defined as identifying if the flights will be delayed or not [21]. For each instance, seven attributes and a target class (delayed or not) are given. The dataset comprises 539,383 samples in total.

Pokerhand: This dataset is composed of 1,025,010 samples, where the task is to predict the poker hand in a set of five cards drawn from a deck of 52 cards. Each hand is described according to 10 attributes including the *suit* and *rank* of each of the five cards in the poker hand. The class attribute $\in [0..9]$ informs the quality of the hand [15].

PKLot Concept Drift: this is a variation of the original PKLot dataset [22]. The test protocol proposed in [9] uses 59 LBP features and defines the classification problem as identifying parking spots as occupied or vacant. In the protocol, at each time step, 50 random samples of each class collected from the previous day are given for training, and all instances from the current day must be classified. The dataset contains 619,466 instances.

V. Evaluation

In [5], [6] it is demonstrated that the data drawn from the Electricity and Forest covertype has a time dependence in the target class. Firstly, we replicate these experiments and extend the results to include all datasets discussed in Section IV.

Since the target class can be seen as a categorical series, we use the Cramér's V measure to compute the serial dependence for the target class (in [5], [6] an autocorrelation function is used since only binary problems are studied). The results are shown in Figure 1. As one can observe in Figure 1, all tested datasets present some serial dependence in the target class attribute. Intuitively, some serial dependence in the target classes for the real-world problems is expected. For instance, the Cramér's V value in the Electricity dataset peaks at every 24 hours, thus indicating that the current price is strongly related to the price from the same hour in previous days.

Note the high Cramér's V value even for large lags in the PKLot dataset. This indicates that this dataset may not be suitable for tests using a test-then-train approach [8], [15] since a blind classifier will achieve high accuracy in this problem. Also, a test-then-train approach in the PKLot dataset is an impractical test, as in the real world, it is unreasonable to acquire the true label from the previous instance before classifying the next one in a parking lot classification problem. The original evaluation protocol proposed in [9] does not use a test-then-train approach for the PKLot. However, the Electricity, Forest, and Pokerhand datasets also show a high serial dependence, and these datasets are often used in a test-then-train fashion [15], [21], [23].

Since in the tested datasets the Cramér's V value peaks when lag $= 1$, in a test-then-train scenario, methods proposed to deal with concept drifts should at least perform better than a blind classifier, that classifies the instance given at t as having the same class from the instance given at $(t-1)$.

Another question that arises in these benchmarks is if there is any concept drift present. Of course, it is a fundamentally difficult question since if we could develop a method to definitely answer this question, this method could be used to create a perfect concept drift detector for any stream. In spite of that, we can use some intuitions in order to suspect that there is some concept drift in the datasets.

Since this is a common practice, all the following tests use the test-then-train approach for all except the PKLot dataset. For the PKLot benchmark, the original protocol discussed in Section IV is used. All results are averaged across 30 runs.

First, since many methods may rely on a trigger to detect and adapt to concept drifts, we test a Hoeffding Tree [24] classifier, that is restarted every time a concept drift is signaled by means of a random trigger, which signals a drift with a probability p for every new training instance received. The rationale in this test is that a random trigger may increase the accuracy if a real concept drift is present, since the discard of previous conflicting data may be beneficial. The random trigger should also be used as a naïve method for comparison with other methods since a trigger that fires at random should perform worse than a well-conceived one. Figure 2 shows the average accuracy achieved by the random trigger according to p in the discussed datasets.

As one can observe in Figure 2, for many of the tested datasets the average accuracy is increased when using a random trigger when $p \in [1, 4]\%$ when compared an incremental learner without any trigger ($p = 0\%$). This indicates that for some problems, a random discard of old data is enough to

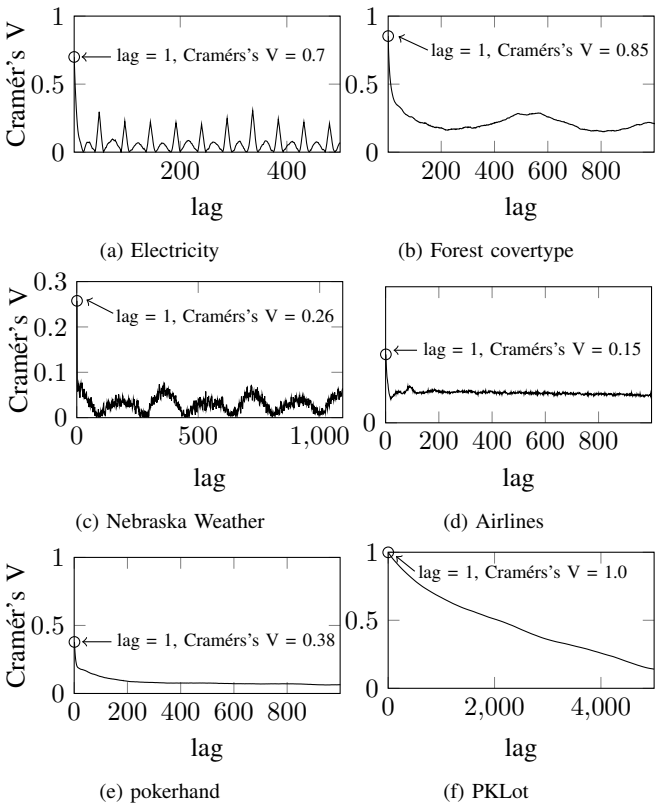


Fig. 1. Target class serial dependence.

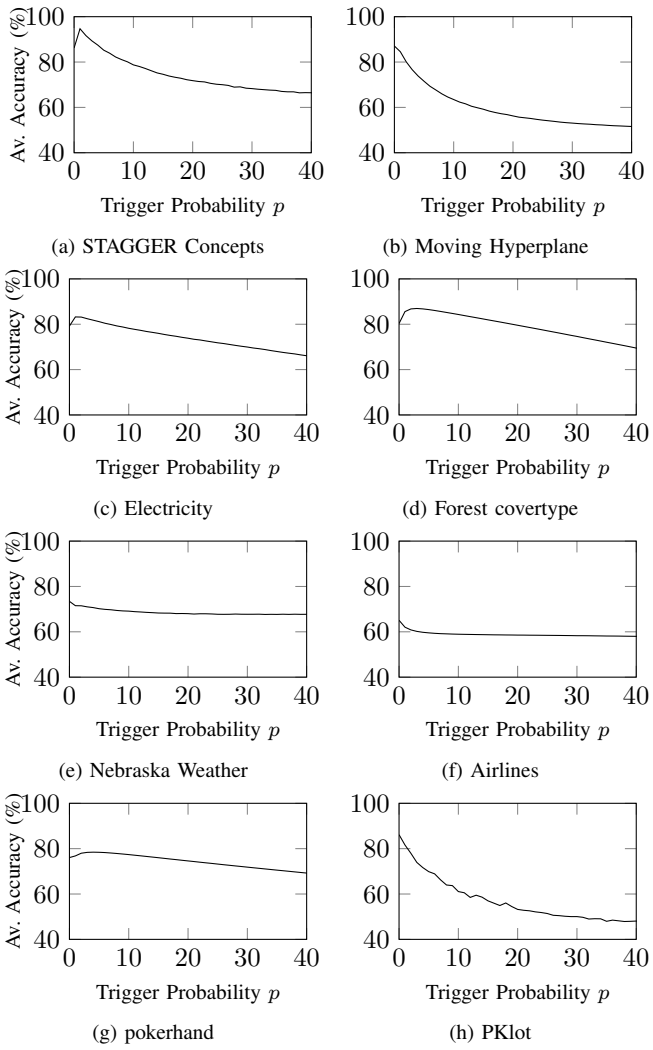


Fig. 2. Average accuracy achieved when using a random trigger.

TABLE I
BEST VALUE OF p USING A RANDOM TRIGGER.

	Hoeffding Tree		Naïve Bayes	
Benchmark	Best p	Avg. Accuracy	Best p	Avg. Accuracy
STAGGER	1%	94.68%	1%	94.74%
Hyperplane	0%	87.06%	1%	85.19%
Electricity	1%	83.23%	2%	83.72%
Forest Cover	3%	86.94%	3%	87.03%
Nebraska	0%	73.43%	2%	71.54%
Airlines	0%	65.08%	0%	64.55%
pokerhand	4%	78.45%	4%	78.68%
PKLot	0%	86.17%	0%	86.1%

increase the average accuracy, and to justify the use of a concept drift detector, methods that deal with concept drifts should generate results at least better than the best random trigger. Table I shows the best value of p and its corresponding average accuracy achieved for the random trigger in each of the tested benchmarks, where we also include the best results obtained using a Naïve Bayes classifier.

Since the main problem caused by a concept drift is the accuracy loss over time [21], one should test a static classifier in the datasets to verify if there is any accuracy drop over time. Since most benchmarks propose finite problems, we suggest to use the first 10% of the data in the stream for training a static (conventional) classifier that should be used as a baseline (naïve approach), and test this classifier in the remaining of the stream (for the static classifier test we ignored the original protocol of the PKLot and used the first 10% samples of the stream for training – i.e., the first 61,946 instances).

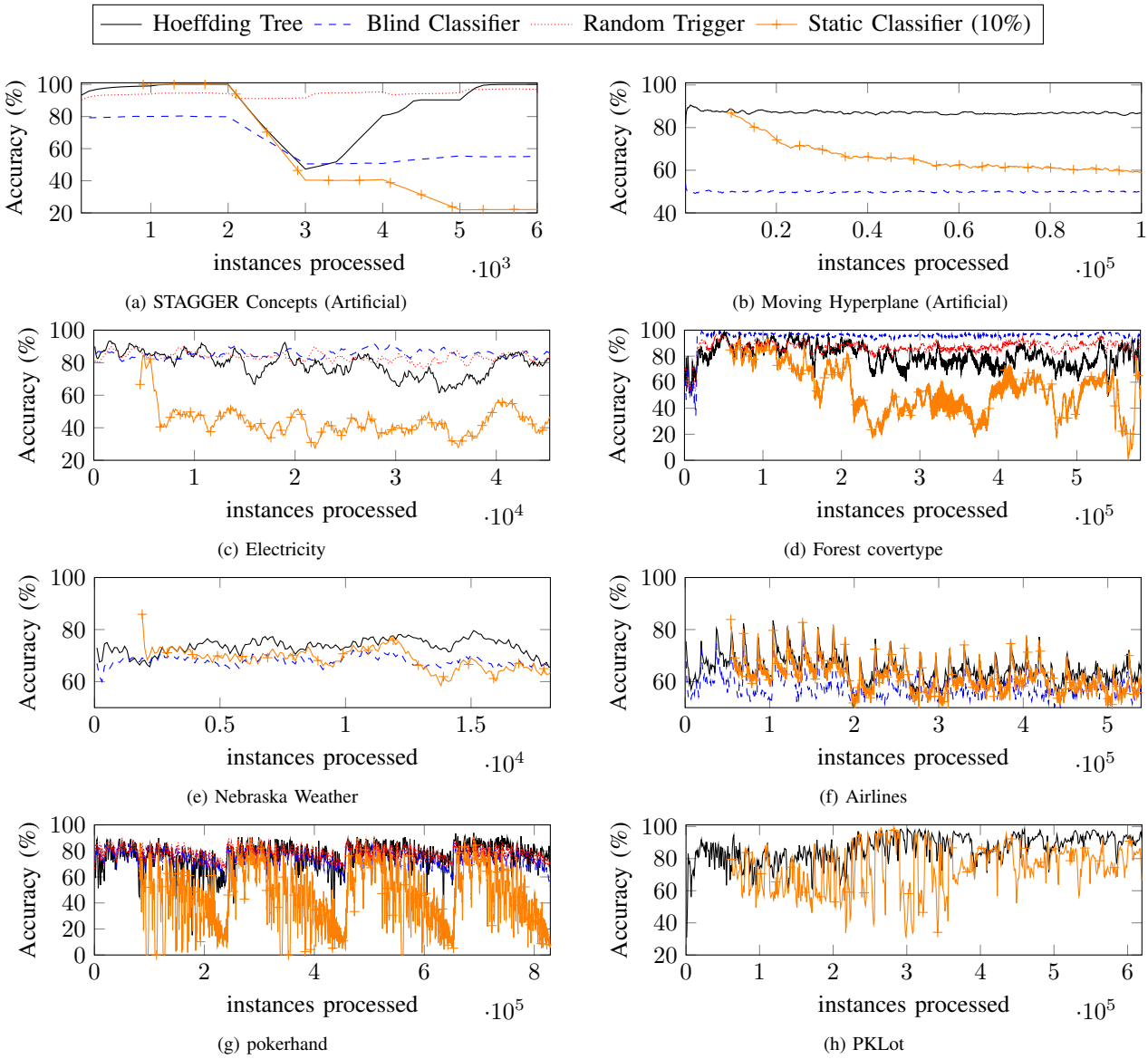
In [21] it is mentioned that a concept drift can be reflected in a decrease in the accuracy over time in incremental learners. Thus, another interesting test that should be made is the use of an incremental learner that does not adapt to concept drifts (e.g., a simple Naïve Bayes or Hoeffding Tree classifier). An incremental learner that does not consider concept drifts should not be able to quickly recover its accuracy after a change, especially under real concept drifts, due to the possibly conflicting data received at different time steps.

Figure 3 shows the prequential accuracy for a window size of 1,000 instances [8] in the real-world datasets to obtain insights about the presence of concept drifts and the classification difficulty level in these datasets. All approaches use Hoeffding Trees as base learners. We also include the results in the STAGGER and Moving Hyperplane artificial datasets for comparison purposes. The random trigger value p for the tests in Figure 3 are the best values for the Hoeffding Tree observed in Table I, and for the benchmarks where the best p is 0, we do not show the Random trigger classifier as it is the same as a Hoeffding Tree classifier without any trigger.

First considering the artificial benchmarks in Figure 3, we can observe that the blind classifier does not perform well (the high accuracy at the beginning of the STAGGER Concepts dataset is given by differences in the *priors*). The static classifier shows severe accuracy losses over time, and the random trigger can keep a stable accuracy over time due to the frequent discard of old data in the STAGGER. When considering the Hoeffding Tree classifier in the STAGGER problem, it is clear that the abrupt changes cause a significant accuracy loss that takes a long time to be mitigated.

The Hoeffding Tree classifier shows a stable prequential accuracy in 3b, which is expected since the concept drift is gradual (a small change for every new instance generated). The good results achieved by the Hoeffding Tree combined with a Random Trigger in Figs. 3a/3b demonstrate that in these simple problems, a random discard of old data, or a classifier that learns in an online fashion may be enough to recover from concept drifts. Nevertheless, it is expected that a method built to deal with concept drifts should surpass the accuracies of these naïve methods.

The results obtained in the real-world benchmarks show some interesting behaviors. As expected, the static classifier does show severe accuracy losses for most tested real-world datasets in Figure 3. On the other hand, the Hoeffding Tree classifier is able to mitigate most of the accuracy drops when compared to the static classifier in Figures 3c, 3d, 3g and 3h, indicating that the drift may be predominantly virtual (i.e., the



(a) STAGGER Concepts (Artificial)

(b) Moving Hyperplane (Artificial)

(c) Electricity

(d) Forest covertype

(e) Nebraska Weather

(f) Airlines

(g) pokerhand

(h) PKLot

Fig. 3. Baseline methods prequential accuracies in the real-world and artificial datasets.

Hoeffding Tree is learning new areas in the feature space as new instances arrive), and/or the changes in the *a posteiori* probabilities are small/gradual in these benchmarks.

The results in the Electricity dataset corroborate with [14], where it is discussed that the drift in this dataset is mostly virtual. Note that for the PKLot dataset in Figure 3h, the static classifier was trained using 61,946 instances, while the Hoeffding Tree was trained in an online fashion using 8,200 samples.

The blind classifier achieved good results in all real-world benchmarks (the result is not available in the PKLot since it does not follow a test-then-train approach)[2], and the best results in the Electricity and Forest covertype datasets. This further indicates the importance of comparing against this classifier, since it is not possible to justify the use of a complex and computationally expensive method when a blind learner may achieve better results. The random trigger classifier was able to keep a stable prequential accuracy, and it achieved the best result in the Pokerhand (Figure 3g) benchmark.

VI. Experiments

Next, we validate the proposed evaluation approach using the DDM [25] trigger using a Hoeffding Tree as the base learner, and the Leveraging Bagging method [26] using the ADWIN [27] as a trigger and a pool of 30 Hoeffding Trees. Figure 4 shows the prequential accuracy over a window of size 1,000 and compares DDM results against the best naïve approach for each dataset discussed in Section V. The average results achieved by all proposed naïve approaches discussed in Section V and the DDM method is available in Table II.

[2]When using a test-then-train approach in the PKLot, the average accuracy achieved is: Hoeffding Tree 94.22%; Blind Classifier 99.96%; Random Trigger 94.22%; DDM 99.87%; LevBag 99.36%; LevRnd 97.48%.

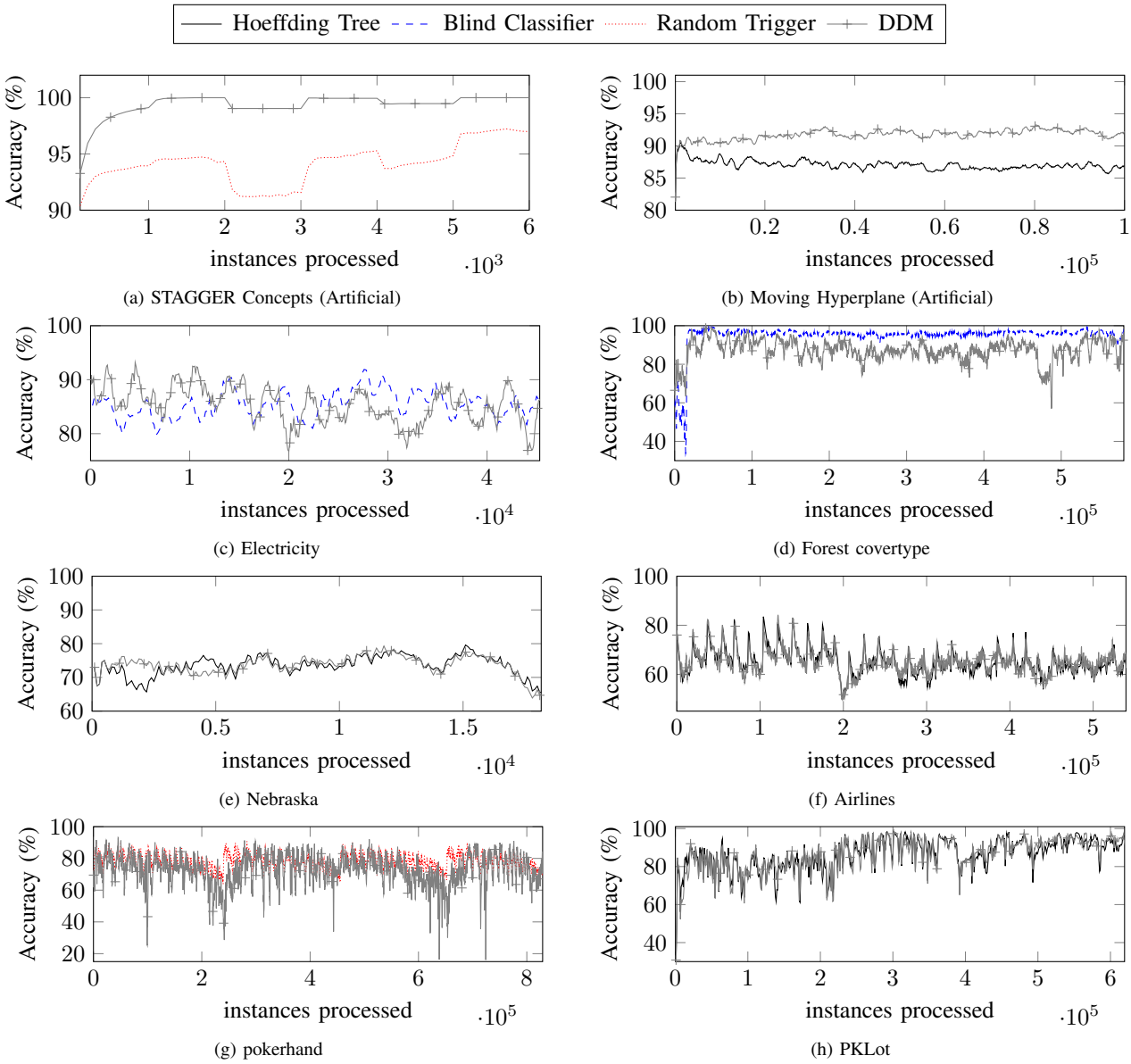


Fig. 4. Experimental results using the DDM method.

The results in Figure 4 and Table II show that the DDM method behaves as expected in a stream that contains concept drifts due to the good results in the artificial benchmarks. However, when analyzing the results obtained in the real-world-benchmarks, it is not clear if the DDM method is indeed adapting to concept drifts. In the Forest Covertype and Pokerhand benchmarks, the method was not able to keep an accuracy higher than some naïve methods.

Figure 5 (only the real-world benchmarks results are presented in Figure 5) and Table III show the result of the tests using the Leveraging Bagging (LevBag) method. The Leveraging Bagging copes with concept drifts employing a pool of classifiers, where the worst performing classifier is replaced when a trigger detects the concept drift. This introduces a problem, as we may not know whether the method is generating better results due to the correct detection and adaptation to concept drifts, or due to the increased number of classifiers and diversity of the pool.

To address this problem, we replaced the original trigger

TABLE II
BASELINE METHODS AND DDM AVERAGE ACCURACY.

Benchmark	Hoeff.	Blind	Random	Static	DDM
STAGGER	86.18%	62.00%	94.68%	46.44%	**99.60%**
Hyperplane	87.06%	49.93%	87.06%	65.64%	**91.76%**
Electricity	79.20%	85.33%	83.23%	43.50%	**85.41%**
Forest	80.31%	**95.06%**	86.94%	53.94%	87.35%
Nebraska	73.43%	68.02%	73.43%	68.50%	**73.64%**
Airlines	65.08%	58.05%	65.08%	61.69%	**65.28%**
Poker	76.06%	74.54%	**78.45%**	44.97%	72.74%
PKLot[2]	86.39%	-	86.39%	73.72%	**86.69%**

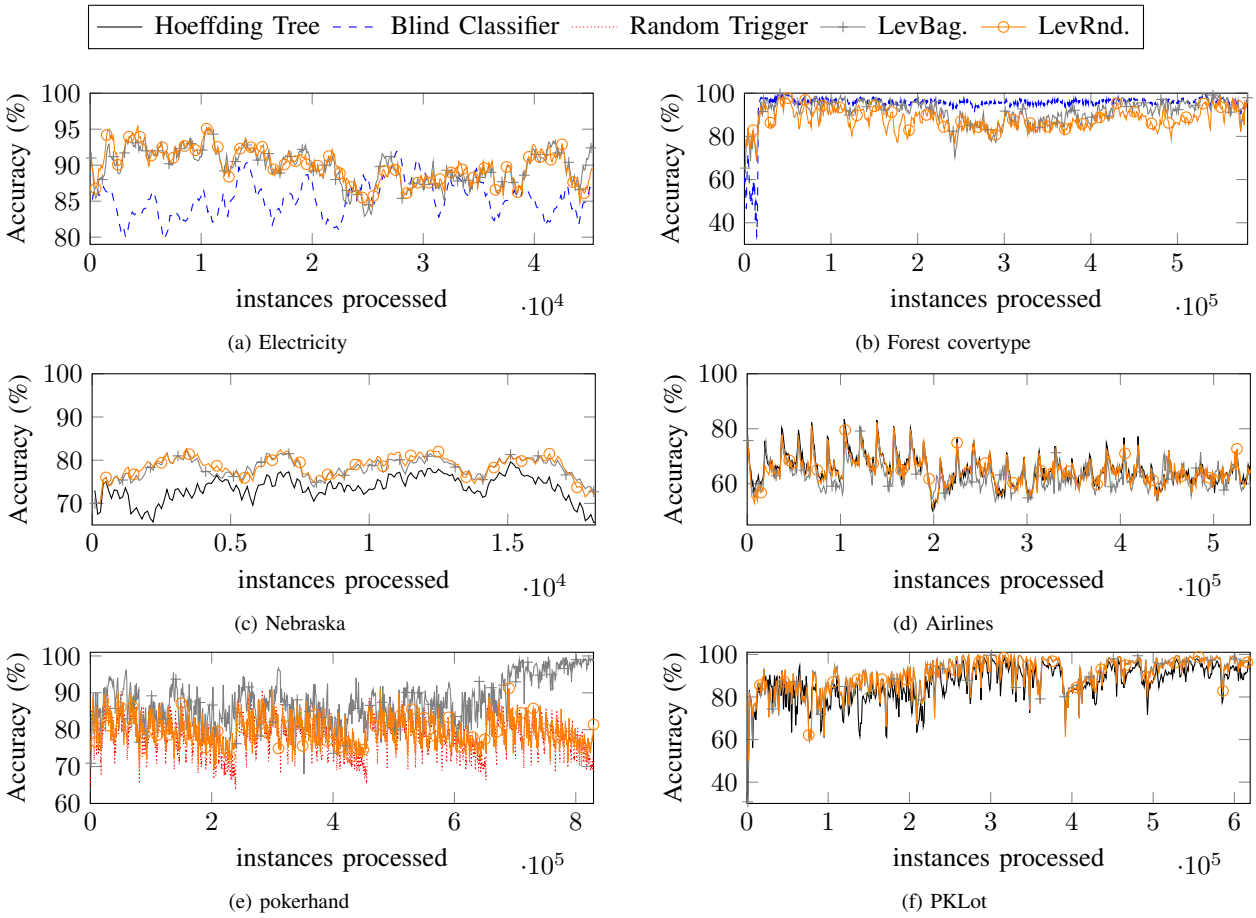


Fig. 5. Experimental results using the Leveraging Bagging method.

TABLE III
BASELINE METHODS AND THE LEVERAGING BAG. AVERAGE ACCURACY.

Benchmark	Hoeff.	Blind	Random	Static	LevBag	LevRnd
STAGGER	86.18%	62.00%	94.68%	46.44%	**97.74%**	97.68%
Hyperplane	87.06%	49.93%	87.06%	65.64%	**93.28%**	89.23%
Electricity	79.20%	85.33%	83.23%	43.50%	89.80%	**89.96%**
Forest	80.31%	**95.06%**	86.94%	53.94%	91.70%	87.88%
Nebraska	73.43%	68.02%	73.43%	68.50%	78.11%	**78.47%**
Airlines	**65.08%**	58.05%	**65.08%**	61.69%	63.14%	64.62%
Poker	76.06%	74.54%	78.45%	44.97%	**87.60%**	79.61%
PKLot[2]	86.39%	-	86.39%	73.72%	**90.36%**	90.08%

used in the method by a Random Trigger, using the p values described in Table I. If the results using the random trigger (LevRnd) are similar to the original method, any accuracy improvement in the benchmarks is probably related to the stronger accuracy of the ensemble.

Beginning with the artificial datasets in Table III, there is an accuracy improvement when using the original Leveraging Bagging method, when compared with the naïve methods in the Moving Hyperplane benchmark. Nevertheless, similar results were achieved when comparing the original versus the Random Trigger version of the method when using the STAGGER Concepts dataset.

When considering the real-world benchmarks in Figure 5 and Table III, note that the results achieved by the original and Random Trigger versions of the Leveraging Bagging method are quite similar for the Electricity, Nebraska and Airlines datasets. According to Table III, the Random Trigger version of the method achieved better results than the original Leveraging Bagging in the Electricity and Nebraska datasets, further indicating that the accuracy improvements are due to the stronger classification abilities of the pool, not to the detection of concept drifts.

In the Forest Covertype dataset, the blind classifier achieved better results than both versions of the Leveraging Bagging method. The same occurs in the Airlines when comparing the Leveraging Bagging to a simple Hoeffding Tree classifier. In the Pokerhand and PKLot datasets, the Leveraging Bagging show an accuracy higher than any naïve method, including the Random Trigger version of the Leveraging Bagging.

It is important to mention that we are not suggesting that the DDM or Leveraging Bagging methods are not able to cope with concept drifts since both are well studied in the literature, and the artificial benchmarks show that these methods can cope with changes. However, the results show that some of the most common benchmarks may not be suitable to put methods that deal with concept drifts to test. The reason is that it is difficult to determine if the methods are indeed adapting to

the concept drifts or if they are triggering at random. This is especially problematic in the test-then-train approach, where clearly some naïve approaches such as a blind classifier or a random trigger can achieve better results, as shown in Figures 4, 5, and in Tables II, III.

VII. Conclusions

In this paper, we showed that some common datasets used for testing techniques tailored for handling drifting data might have only minor changes, if any. Therefore, we proposed simple naïve approaches for benchmarking, where methods that deal with concept drifts should have their accuracy rates compared with such approaches. Experimental results show that the naïve approaches can be better than some classical methods to deal with possible concept drifts in datasets such as the Forest Covertype, Electricity, and Nebraska Weather, thus suggesting that these benchmarks may present trivial problems in the concept drift viewpoint. This indicates that more robust datasets for testing concepts drift approaches are needed.

Besides being a common practice that is reproducible in the real-world for some problems, we also argue that a test-then-train approach is an unrealistic test under some scenarios. For instance, how can we assume a test instance will be immediately labeled after the test in the Forest Covertype problem? For these problems, we recommend to not consider the entire stream as labeled, or insert some delay when labeling the instances [8].

The results show the importance of using a range of datasets to put methods that deal with concept drifts to test, including artificial problems, where the properties of the concept drifts are known. Also, the use of naïve methods can be of special importance to researchers that wish to propose new datasets containing concept drifts since, in these scenarios, it is imperative to give some evidence that a concept drift is present in the proposed problem.

As future work, we intend to expand the analysis to new benchmarks in order to find more representative concept drift problems and to consider new naïve methods and metrics to give better insights of the quality of the benchmarks for assessing methods that deal with concept drifts.

References

[1] J. Schlimmer and R. Granger, Jr., "Incremental learning from noisy data," *Machine Learning*, vol. 1, no. 3, pp. 317–354, 1986.

[2] G. Hulten, L. Spencer, and P. Domingos, "Mining time-changing data streams," in *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '01. ACM, 2001, pp. 97–106.

[3] H. M. Gomes, J. Read, A. Bifet, J. P. Barddal, and J. Gama, "Machine learning for streaming data: state of the art, challenges, and opportunities," *SIGKDD Explorations*, vol. 21, no. 2, pp. 6–22, 2019. [Online]. Available: https://doi.org/10.1145/3373464.3373470

[4] S. Wares, J. Isaacs, and E. Elyan, "Data stream mining: methods and challenges for handling concept drift," *SN Applied Sciences*, vol. 1, no. 11, p. 1412, 2019.

[5] A. Bifet, J. Read, I. Žliobaitė, B. Pfahringer, and G. Holmes, "Pitfalls in benchmarking data stream classification and how to avoid them," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2013, pp. 465–479.

[6] A. Bifet, "Classifier concept drift detection and the illusion of progress," in *International Conference on Artificial Intelligence and Soft Computing*. Springer, 2017, pp. 715–725.

[7] T. R. Hoens, R. Polikar, and N. V. Chawla, "Learning from streaming data with concept drift and imbalance: an overview," *Progress in Artificial Intelligence*, vol. 1, no. 1, pp. 89–101, 2012.

[8] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM computing surveys (CSUR)*, vol. 46, no. 4, p. 44, 2014.

[9] P. R. Almeida, L. S. Oliveira, A. S. Britto Jr, and R. Sabourin, "Adapting dynamic classifier selection for concept drift," *Expert Systems with Applications*, vol. 104, pp. 67–85, 2018.

[10] C. H. Weiß and R. Göb, "Measuring serial dependence in categorical time series," *AStA Advances in Statistical Analysis*, vol. 92, no. 1, pp. 71–89, 2008.

[11] C. Weiss, *An Introduction to Discrete-Valued Time Series*. Wiley, 2018.

[12] I. Zliobaite, "How good is the electricity benchmark for evaluating concept drift adaptation. corr abs/1301.3524 (2013)," 2013.

[13] G. I. Webb, R. Hyde, H. Cao, H. L. Nguyen, and F. Petitjean, "Characterizing concept drift," *Data Mining and Knowledge Discovery*, vol. 30, no. 4, pp. 964–994, 2016.

[14] G. I. Webb, L. K. Lee, B. Goethals, and F. Petitjean, "Analyzing concept drift and shift from sample data," *Data Mining and Knowledge Discovery*, vol. 32, no. 5, pp. 1179–1199, 2018.

[15] A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, and R. Gavaldà, "New ensemble methods for evolving data streams," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '09. New York, NY, USA: ACM, 2009, pp. 139–148.

[16] M. Harries, "Splice-2 comparative evaluation: Electricity pricing," University of New South Wales, Tech. Rep., 1999.

[17] J. a. Gama, R. Rocha, and P. Medas, "Accurate decision trees for mining high-speed data streams," in *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '03. ACM, 2003, pp. 523–528.

[18] R. Elwell and R. Polikar, "Incremental learning of concept drift in nonstationary environments," *Neural Networks, IEEE Transactions on*, vol. 22, no. 10, pp. 1517–1531, Oct 2011.

[19] T. Escovedo, A. V. A. Da Cruz, M. M. B. R. Vellasco, and A. S. Koshiyama, "Learning under concept drift using a neuro-evolutionary ensemble," *International Journal of Computational Intelligence and Applications*, vol. 12, no. 04, p. 1340002, 2013.

[20] E. Ikonomovska, J. Gama, and S. Džeroski, "Learning model trees from evolving data streams," *Data mining and knowledge discovery*, vol. 23, no. 1, pp. 128–168, 2011.

[21] J. Demšar and Z. Bosnić, "Detecting concept drift in data streams using model explanation," *Expert Systems with Applications*, vol. 92, pp. 546–559, 2018.

[22] P. R. De Almeida, L. S. Oliveira, A. S. Britto Jr, E. J. Silva Jr, and A. L. Koerich, "Pklot–a robust dataset for parking lot classification," *Expert Systems with Applications*, vol. 42, no. 11, pp. 4937–4949, 2015.

[23] A. Pesaranghader and H. L. Viktor, "Fast hoeffding drift detection method for evolving data streams," in *Joint European conference on machine learning and knowledge discovery in databases*. Springer, 2016, pp. 96–111.

[24] P. Domingos and G. Hulten, "Mining high-speed data streams," in *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '00. New York, NY, USA: Association for Computing Machinery, 2000, p. 71–80.

[25] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, "Learning with drift detection," in *Brazilian symposium on artificial intelligence*. Springer, 2004, pp. 286–295.

[26] A. Bifet, G. Holmes, and B. Pfahringer, "Leveraging bagging for evolving data streams," in *Joint European conference on machine learning and knowledge discovery in databases*. Springer, 2010, pp. 135–150.

[27] A. Bifet and R. Gavaldà, "Learning from time-changing data with adaptive windowing," vol. 7, 04 2007.