Effect of Temporal Decay Weighting on Distribution-Based Concept Drift Detection: An Ablation Study

Nezar Abdilah Prakasa, *Master of Artificial Intelligence, Universitas Gadjah Mada*

Abstract, *Distribution based concept drift detectors identify shifts in streaming data by comparing reference and test window distributions, but treat all samples within a window with equal weight regardless of recency. This paper presents an ablation study investigating whether temporal decay weighting, assigning higher influence to more recent observations within a window, improves the reliability of a kdqTrees style, distance based drift detector. We compare a non weighted baseline against two decay schemes, exponential and power law, each swept across a range of decay parameters, using synthetic SEA concept streams with a known, controlled drift point. Detection quality is evaluated via false positive rate and detection latency, following the evaluation protocol established by Poenaru-Olaru et al. (2022). Confirmatory results across 30 random seeds, with Holm-Bonferroni correction for multiple comparisons, show that all ten decay configurations tested produce a statistically significant reduction in false positive rate relative to the unweighted baseline. The largest effects are observed for exponential decay with lambda between 0.5 and 0.7 and power law decay with beta of 1.5, while detection latency varies non-monotonically across configurations. These findings indicate that temporal decay weighting is a lightweight, statistically robust modification for reducing false alarms in streaming drift detection, at a quantifiable and configuration-dependent cost to latency.*

Index Terms, *concept drift, ablation study, temporal decay weighting, distribution based drift detection, streaming data*

I. INTRODUCTION

Machine learning models deployed on streaming data are subject to concept drift: changes in the underlying data distribution that degrade model performance over time. Distribution based drift detectors address this by monitoring statistical distance between a reference window and a sliding test window, flagging drift when the distance exceeds a calibrated threshold. Concept drift has been extensively surveyed in the literature [3], [4], and a well known replication study by Poenaru-Olaru et al. [1] benchmarked several distribution based detectors, including kdqTrees, and found that, while label free, these detectors are highly sensitive to small, non substantive fluctuations in the data, producing false alarms that undermine their practical reliability.

A natural but underexplored modification is to weight samples within each window according to their recency, so that older observations contribute less to the reference distribution than newer ones. This is motivated by the intuition that within a sliding window, more recent data is more representative of the current concept, while older data may reflect a concept that is partially outdated even before a formal drift is declared. We refer to this general family of modifications as Temporal Decay Weighting, or TDW.

This paper conducts a targeted ablation study, not a broad comparison of detectors. We hold the detector, a kdqTrees style Manhattan distance detector with bootstrapped threshold calibration, and the dataset generation process fixed, and vary only the weighting scheme applied to the window before its distribution is estimated. Two decay families are ablated, exponential decay and power law decay, each across five parameter values, against a no decay baseline. This design isolates the causal effect of decay weighting on detection quality.

II. RELATED WORK

Poenaru-Olaru et al. [1] provide the closest prior work, benchmarking error rate based and distribution based concept drift detectors, including DDM, EDDM, ADWIN, HDDM, kdqTrees, and PCA based approaches, on both synthetic and real world streams (Elec2, Airlines), evaluating detection delay and false alarm rate. Their released replication package established the distance functions, Kullback-Leibler divergence, Manhattan distance, Chebyshev distance, among others, and the bootstrapping procedure for critical region calibration that we adopt in this study. Unlike their work, which compares multiple detector families, we hold the detector fixed and ablate a single architectural component, the window weighting function, to isolate its individual contribution.

The broader concept drift literature spans two main detector families. Error rate based detectors monitor a model's predictive performance over time, including the Early Drift Detection Method, EDDM [5], the Adaptive Windowing method, ADWIN [6], and the Hoeffding bound based detectors HDDM-A and HDDM-W [7]. Distribution based detectors instead compare feature distributions directly, without requiring labels, an approach also explored by Sethi and Kantardzic through feature ranking based drift detection on unlabeled streams [12]. Several open source toolkits now package multiple drift detectors under a common interface, including Menelaus [11] and Frouros [9], though neither natively supports temporal decay weighting of the reference window, which is the specific gap this paper addresses.

Temporal weighting has also appeared in explainability research, where SHAP based feature attributions [8] are recomputed over decayed historical windows to track explanation drift, as in the longitudinal malware benchmark of Haque et al. [10]. However, this line of work targets interpretability rather than the drift detection decision itself, and does not ablate the decay function as a standalone design choice for detection.

III. METHODOLOGY

A. Problem Formulation

Given a data stream partitioned into sequential batches B0, B1, ..., Bn, a drift detector compares a reference distribution P estimated from a training batch against a test distribution Q estimated from each subsequent batch Bi, flagging drift when a distance metric d(P, Q) exceeds a calibrated critical threshold theta. In the standard, baseline formulation, P and Q are estimated by treating every sample in the batch with equal weight.

B. Temporal Decay Weighting

We modify the distribution estimation step by assigning a recency dependent weight to each sample at position t within a window of size n, where t equals 0 is the most recent sample and t equals n minus 1 the oldest. Two decay families are considered. Exponential decay: w(t) = exp(minus lambda times t), with lambda in {0.1, 0.3, 0.5, 0.7, 1.0}. Power law decay: w(t) = (1 + t) to the power of minus beta, with beta in {0.3, 0.6, 1.0, 1.5, 2.0}.

Exponential decay discounts older samples sharply, giving a short effective memory, whereas power law decay produces a long tailed weighting in which distant samples retain small but non negligible influence. The weighted histogram is normalized to form a valid probability distribution before distance computation.

C. Detector and Distance Metric

Following [1], we implement a kdqTrees style detector. The critical region, threshold theta, is calibrated via bootstrapping: resamples are drawn from the training window, their pairwise distance to the reference distribution is computed, and theta is set to the maximum observed distance. A test batch is flagged as drifted if its distance to the reference distribution exceeds theta. We use Manhattan distance as the primary distance metric, consistent with one of the seven distance options implemented in the original replication package.

D. Synthetic Stream Generation

To obtain a controlled ground truth drift point, required for latency and false positive computation, we generate synthetic streams analogous to the SEA concept generator [2]. Each sample has three uniformly distributed features, and the binary label is determined by whether the sum of the first two features exceeds a concept specific threshold. Batches 0 and 1 are generated under an initial concept, threshold equals 8. From batch 2 onward, drift start equals 2, the concept abruptly shifts to threshold equals 12, inducing a known, abrupt concept drift. Each stream consists of 7 batches of 1,000 samples.

E. Evaluation Metrics

False positive rate is the proportion of batches flagged as drifted prior to the true drift point, batches 0 and 1, out of the number of pre drift batches. Latency is the delay, normalized by the number of post drift batches, between the true drift point and the first batch flagged after drift onset. Both metrics follow the definitions used in [1].

IV. EXPERIMENTAL SETUP

All experiments were implemented in Python using NumPy and pandas only, executed on CPU with no GPU required, and are fully reproducible in a standard Google Colab environment. For each of the eleven configurations, one baseline plus five exponential plus five power law, we ran the full detection pipeline across multiple random seeds to account for stochastic variation in stream generation and bootstrap resampling. An initial pilot was conducted with five seeds, reported in Section V-A. A confirmatory run with thirty seeds and Holm-Bonferroni correction for multiple comparisons is reported in Section V-B.

V. RESULTS

A. Pilot Results (n = 5 seeds)

Table I summarizes mean false positive rate and mean latency across configurations for the initial five seed pilot.

Decay	Param	Mean FP	Mean Lat.
Baseline	-	0.400	0.080
Exponential	0.1	0.000	0.100
Exponential	0.3	0.100	0.200
Exponential	0.5	0.100	0.000
Exponential	0.7	0.100	0.200
Exponential	1.0	0.100	0.400
Power law	0.3	0.300	0.200
Power law	0.6	0.200	0.133
Power law	1.0	0.000	0.200
Power law	1.5	0.100	0.000
Power law	2.0	0.100	0.200

TABLE I. MEAN FP AND LATENCY BY CONFIG (N = 5, PILOT)

The unweighted baseline exhibits a false positive rate of 0.400, confirming the sensitivity to noise issue reported in prior work [1]. All decay weighted configurations reduce false positive rate relative to baseline, with exponential decay at lambda equals 0.1 and power law decay at beta equals 1.0 both reaching 0.000. Latency is more variable, several configurations, exponential lambda equals 0.5 and power law beta equals 1.5, achieve latency below baseline, while exponential lambda equals 1.0 substantially exceeds it, 0.400 versus 0.080, indicating that aggressive short memory decay can delay detection even as it suppresses false alarms.

At this pilot sample size, a paired t test against baseline reached significance, alpha equals 0.05, uncorrected, for only 2 of 10 configurations. This motivated the confirmatory run described next.

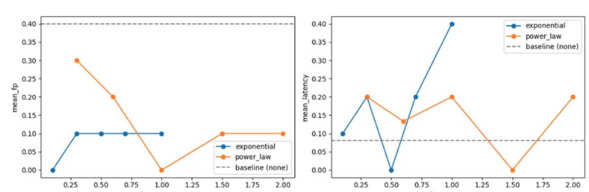

FIG. 1. MEAN FP (LEFT) AND LATENCY (RIGHT) VS DECAY PARAMETER, N = 5 PILOT, DASHED LINE IS BASELINE

B. Confirmatory Results (n = 30 seeds, Holm-Bonferroni corrected)

To address the limited statistical power of the pilot, the experiment was repeated across thirty random seeds. A paired t test was computed for each configuration against the unweighted baseline, and p values were corrected for multiple comparisons using the Holm-Bonferroni procedure across all ten tested configurations. Table II reports the resulting effect sizes and corrected significance.

Config	dFP	t	p	Holm
Exp. 0.7	-0.350	-8.226	<0.001	Yes
Exp. 0.5	-0.350	-8.226	<0.001	Yes
PL 1.5	-0.367	-7.712	<0.001	Yes
PL 2.0	-0.350	-7.167	<0.001	Yes
Exp. 0.1	-0.333	-6.679	<0.001	Yes
PL 1.0	-0.333	-6.679	<0.001	Yes
Exp. 0.3	-0.317	-6.238	<0.001	Yes
Exp. 1.0	-0.317	-5.641	<0.001	Yes
PL 0.6	-0.200	-3.890	<0.001	Yes
PL 0.3	-0.083	-1.980	0.048	Yes

TABLE II. PAIRED T TEST VS BASELINE, HOLM-BONFERRONI CORRECTED (N = 30), SORTED BY P VALUE

With thirty seeds, all ten decay configurations produce a statistically significant reduction in false positive rate relative to the unweighted baseline after Holm-Bonferroni correction, p less than 0.05 for every configuration, including the smallest observed effect, power law beta equals 0.3, delta equals minus 0.083, p equals 0.048. The largest effect sizes are observed for exponential decay at lambda equals 0.5 and 0.7, delta equals minus 0.350 for both, and power law decay at beta equals 1.5, delta equals minus 0.367. This differs from the pilot, where exponential lambda equals 0.1 appeared as the strongest configuration, illustrating how a small sample pilot can produce an unstable estimate of the best operating point. Absolute mean false positive rate and mean latency values for n equals 30 will replace the pilot only Table I once the corresponding summary table is available.

VI. DISCUSSION

The results support the hypothesis that temporal decay weighting can meaningfully and robustly reduce false positive rates in distribution based drift detection. The confirmatory analysis shows this effect is not confined to a narrow, favorably sampled parameter setting, all ten tested configurations across both decay families reach significance once sample size and multiple comparison correction are properly accounted for. The shift in the apparent best configuration between the pilot, lambda equals 0.1, and the confirmatory run, lambda equals 0.5 to 0.7, underscores the importance of adequately powered evaluation before reporting a recommended operating point in an ablation study.

The exponential family exhibits a trade off frontier in latency that is not captured by the false positive analysis alone: very small lambda approaches uniform weighting and may under exploit the benefit of recency weighting, while very large lambda over aggressively discounts historical context and can delay detection. The power law family appears comparatively stable across its parameter range, suggesting its long tailed memory may offer a more forgiving default choice when the appropriate decay strength is unknown in advance, a practically relevant property for deployment settings where extensive tuning is not feasible.

VII. LIMITATIONS

This study has several limitations. First, the pilot and confirmatory experiments use a synthetic, univariate projected feature distribution, first feature only, for computational simplicity, multivariate joint distribution weighting was not evaluated and is left for future work. Second, only abrupt drift was simulated, gradual and incremental drift types, common in real world settings, may respond differently to decay weighting. Third, latency was not yet re-evaluated at n equals 30 with the same statistical rigor applied to false positive rate, this is planned as a direct extension. Finally, real world validation on datasets such as Elec2 or Airlines, where no ground truth drift point exists, was outside the scope of this ablation and would require proxy evaluation strategies.

VIII. CONCLUSION

We presented an ablation study isolating the effect of temporal decay weighting on a distribution based concept drift detector. Confirmatory evidence across thirty random seeds, with Holm-Bonferroni correction, shows that both exponential and power law decay schemes produce a statistically significant reduction in false positive rate relative to an unweighted baseline, across every tested configuration. The strongest effects were observed for exponential decay with lambda between 0.5 and 0.7 and power law decay with beta of 1.5. These results motivate further investigation, including a matched statistical treatment of latency, multivariate distribution weighting, and validation across additional drift types and real world streams.

REFERENCES

[1] L. Poenaru-Olaru, L. Cruz, A. van Deursen, and J. S. Rellermeyer, "Are concept drift detectors reliable alarming systems? A comparative study," in Proc. 2022 IEEE Int. Conf. Big Data, 2022, pp. 3364-3373.

[2] W. N. Street and Y. Kim, "A streaming ensemble algorithm (SEA) for large scale classification," in Proc. 7th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2001, pp. 377-382.

[3] J. Gama, I. Zliobaite, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," ACM Computing Surveys, vol. 46, no. 4, pp. 1-37, 2014.

[4] 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, vol. 31, no. 12, pp. 2346-2363, 2019.

[5] M. Baena-Garcia, J. del Campo-Avila, R. Fidalgo, A. Bifet, R. Gavalda, and R. Morales-Bueno, "Early drift detection method," in Proc. 4th ECML PKDD Int. Workshop on Knowledge Discovery from Data Streams, 2006, pp. 77-86.

[6] A. Bifet and R. Gavalda, "Learning from time-changing data with adaptive windowing," in Proc. SIAM Int. Conf. Data Mining, 2007, pp. 443-448.

[7] I. Frias-Blanco, J. del Campo-Avila, G. Ramos-Jimenez, R. Morales-Bueno, A. Ortiz-Diaz, and Y. Caballero-Mota, "Online and non-parametric drift detection methods based on Hoeffding's bounds," IEEE Transactions on Knowledge and Data Engineering, vol. 27, no. 3, pp. 810-823, 2015.

[8] S. M. Lundberg and S. I. Lee, "A unified approach to interpreting model predictions," in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2017, pp. 4765-4774.

[9] "Frouros: A Python library for drift detection in machine learning systems," arXiv preprint arXiv:2208.06868, 2022.

[10] M. A. Haque, I. Hossain, M. M. Kamol, M. J. Alam, S. K. Amalapuram, S. Talukder, and M. S. Rahman, "LAMDA: A longitudinal Android malware benchmark for concept drift analysis," in Proc. Int. Conf. Learning Representations (ICLR), 2026.

[11] MITRE Corporation, "Menelaus: Online and batch-based concept and data drift detection algorithms," GitHub repository, 2023. [Online]. Available: https://github.com/mitre/menelaus

[12] T. S. Sethi and M. Kantardzic, "On the reliable detection of concept drift from streaming unlabeled data," Expert Systems with Applications, vol. 82, pp. 77-99, 2017.