

Buffer-Free Real-Time Forecasting Under Concept Drift

Manivannan B¹, Roshini B²

¹Assistant Professor, Department of Computer Science and Engineering, Vivekanandha College of Engineering for Women, Tamil Nadu, India

²PG Scholar, Department of Computer Science and Engineering, Vivekanandha College of Engineering for Women, Tamil Nadu, India

Abstract—Real-time forecasting systems in energy, transport, finance and industrial monitoring work on data streams whose statistical properties change over time, a problem known as concept drift. A forecasting model trained on past data loses accuracy when drift occurs, so it must notice the change and adapt. Existing drift-handling methods usually keep a sliding window, a replay buffer, a memory module or an ensemble of models to detect drift and update the forecaster. These designs increase memory use and delay, are hard to run on edge devices, and require raw samples to be stored. This paper surveys representative work on drift detection, drift-aware ensembles, input normalisation, online time series forecasting and online learning, compares their datasets, mechanisms and memory demands, and identifies open problems: dependence on windows, buffers or several models, slow reaction to gradual drift, the difficulty of choosing window sizes, and evaluation that rarely reports memory and latency. Based on the survey, we outline a conceptual buffer-free framework that normalises the stream with running statistics, forecasts with a single incremental model, detects drift from the prediction error and adapts the model using only the current observation. The framework is a design proposal. It has not been implemented or evaluated here, and no performance claims are made for it.

Index Terms—Concept Drift, Real-Time Forecasting, Buffer-Free Learning, Online Learning, Data Streams, Drift Detection, Time Series.

I. INTRODUCTION

Forecasting the next values of a time series from continuously arriving data is needed in electricity load management, traffic control, financial monitoring and industrial Internet of Things (IoT) sensing. In these settings data is not a fixed dataset. It arrives as a stream, one observation at a time, and the forecast

must be ready before the next observation arrives. A forecasting model therefore cannot rely on many passes over stored history. It has to learn on the fly, answer quickly and use limited memory.

Data streams are rarely stationary. Seasons, policy changes, sensor ageing, sudden events and changes in user behaviour alter the relation between past and future values. This change in the underlying distribution is called concept drift [1], [2]. When drift occurs, the patterns a model has learned stop being valid and the forecast error rises. Drift can be sudden, gradual, incremental or recurring, and a useful forecaster has to handle all of these with little delay.

The usual defence is to monitor the stream with a drift detector and to retrain or fine-tune the model on recently stored samples. Detectors such as DDM [4] and ADWIN [3] watch the error rate or an adaptive window, ensembles such as Adaptive Random Forest [5] attach a detector to every learner, and recent deep online forecasters add memory modules [8] or combine several models [9]. These designs work, but their memory and computation grow with the window length, the size of the memory module or the number of models, and they keep raw data in buffers, which raises storage and privacy concerns.

This survey is concerned with a buffer-free alternative. A buffer-free forecaster processes each observation once, keeps only a small set of running statistics and the parameters of one lightweight model, detects drift from its own prediction error, and adapts immediately. Online learning theory [10] shows that sequential updates with constant memory are possible, and recent online forecasters [8], [9], [11] show that drift-aware forecasting is achievable, but we did not find in the work reviewed here a method that combines drift detection and online adaptation for forecasting without a window, buffer or ensemble.

This paper reviews representative literature and outlines a conceptual framework, which we call the Buffer-Free Drift-Aware Forecasting framework. Section II gives background, Section III reviews the studies, Section IV outlines the framework, Section V compares the reviewed work and proposes an evaluation protocol, Section VI analyses the findings and open problems, and Section VII concludes.

II. BACKGROUND

A. Concept Drift in Streaming Forecasting

Concept drift means that the joint distribution of inputs and target changes over time. In forecasting it appears as a change in the relation between recent history and the next value. Gama et al. [1] describe four common patterns. In sudden drift a new concept replaces the old one at once, for example when a sensor is replaced. In gradual drift the old and new concepts alternate for a while. In incremental drift the data moves slowly and continuously from one concept to another, and in recurring drift an earlier concept returns, as with seasons. Drift is also divided into real drift, where the relation between input and target changes, and virtual drift, where only the input distribution changes [2]. Real drift reduces accuracy directly. Virtual drift can often be reduced by normalising the input. In a stream, the true value becomes known only after the forecast horizon, so a drift detector based on the forecast error receives its signal with a delay of at least one horizon.

B. Building Blocks

Windows and buffers. A sliding window keeps the most recent samples for detection or retraining. A short window reacts fast but gives noisy detection, and a long window is stable but reacts late, so the window size must be tuned for each stream [1], [3]. Error-based detection. DDM [4] tracks the online error rate and raises a warning level and then a drift level when the error rises beyond statistical limits, and ADWIN [3] compares the means of two sub-windows. Ensembles. Adaptive Random Forest [5] and OneNet [9] keep several learners and replace or re-weight them according to recent performance. Input normalisation. RevIN [6] and Dish-TS [7] normalise each input window with its own statistics, or with learned coefficients, and restore the scale at the output, which reduces the effect of shifts in level and scale. Online learning. An online learner updates its parameters after

every observation, usually with one gradient step, so its memory does not depend on the length of the stream [10]. Prequential evaluation. Each observation is first used to test the model and then to train it, which is the standard way to evaluate streaming learners [1].

C. Datasets

Online and drift-aware forecasting studies commonly use public multivariate time series such as the ETT (Electricity Transformer Temperature) datasets, the Electricity consumption dataset, the Weather dataset, the Traffic dataset and the Exchange-rate dataset, as seen in the online forecasting benchmarks of [8] and [9]. Synthetic streams with known drift points are also used because they allow the detection delay to be measured exactly [1], [3]. These datasets differ in length, sampling interval and drift behaviour, so reported errors are not directly comparable across studies. The final choice of datasets for the framework outlined here is left to the implementation phase.

III. LITERATURE REVIEW

The studies below were selected for their work on drift adaptation, drift detection, drift-aware forecasting and online learning. Their descriptions are taken from the papers' abstracts and have not been reproduced here, and no result is quoted unless it appears in the abstract.

A. Drift Surveys and Classical Detectors

[1] Gama et al., in *A Survey on Concept Drift Adaptation*, organise drift adaptation for learning from streams. The survey defines the types of drift, groups adaptation methods by how they remember, detect change, learn and estimate loss, and describes evaluation practice for streams. It is the standard reference for the field, but it is a review, and most of the methods it covers depend on windows or stored samples. [2] Lu et al., in *Learning under Concept Drift: A Review*, present a framework of drift detection, drift understanding and drift adaptation, and summarise datasets and open directions. The review helps in choosing a strategy, but it proposes no algorithm, and the adaptation strategies it describes mostly rely on stored windows or ensembles.

[3] Bifet and Gavaldà, in *Learning from Time-Changing Data with Adaptive Windowing*, introduce ADWIN. It keeps a window of variable length and drops the older part when two sub-windows have

significantly different means, and the authors give guarantees on false positive and false negative rates. It needs the window to be stored, so memory grows when the stream is stable, and it was designed for detecting change in a stream or a classifier's error, not for forecasting. [4] Gama et al., in *Learning with Drift Detection*, introduce DDM. It tracks the online error rate and its deviation and raises a warning and then a drift level when the error rises beyond statistical limits. Detection needs very little memory, but the method is designed for classification, tends to react slowly to gradual drift, and rebuilds the model from the samples gathered after the warning.

B. Ensemble and normalisation methods

[5] Gomes et al., in *Adaptive Random Forests for Evolving Data Stream Classification*, extend the random forest with online bagging, random feature subsets and a drift detector for every tree. When a warning is raised for a tree, a background tree is trained in parallel and replaces it if drift is confirmed. The method recovers well after drift, but many trees and background learners must be kept at the same time, and it is designed for classification. [6] Kim et al., in *Reversible Instance Normalization for Accurate Time-Series Forecasting against Distribution Shift*, normalise each input sequence with its own mean and standard deviation and de-normalise the model output with the same statistics. The method is simple and can be added to existing forecasting models, but it handles only shifts in input statistics. It neither detects drift nor changes the model. [7] Fan et al., in *Dish-TS: A General Paradigm for Alleviating Distribution Shift in Time Series Forecasting*, separate the shift into a part between the input window and the forecast horizon and a part inside the input window, and learn coefficients with two networks to normalise and restore the data. The approach is trained offline on look-back windows, so the model is not updated as the stream arrives, and it adds networks that must be trained and stored.

C. Online drift-aware forecasters

[8] Pham et al., in *Learning Fast and Slow for Online Time Series Forecasting*, propose FSNet, inspired by complementary learning systems. A per-layer adapter supports fast learning of recent changes, and an associative memory supports remembering, updating and recalling repeating events, so that both abrupt and

recurring patterns are handled. The authors report efficacy on real and synthetic datasets. The memory module and adapters add storage and complexity, and the network is updated from recent samples.

[9] Zhang et al., in *OneNet: Enhancing Time Series Forecasting Models under Concept Drift by Online Ensembling*, dynamically update and combine two models, one for dependency across time and one for dependency across variables, with weights adjusted by a reinforcement-learning-based approach inside an online convex programming framework. The authors report a reduction of online forecasting error by more than 50% against the previous best method. Two models must be stored and run for every prediction, so memory and latency are higher than for a single model. [11] Zhao and Shen, in *Proactive Model Adaptation Against Concept Drift for Online Time Series Forecasting*, present Proceed. It estimates the concept drift between recently used training samples and the current test sample, and an adaptation generator turns the estimate into parameter adjustments, so the model is adapted before it forecasts. The generator is trained on synthetic drifts, and the authors report larger gains than earlier online learning methods on five real datasets. The method still works from recently used samples and needs the extra generator to be trained beforehand.

D. Online learning foundations

[10] Hoi et al., in *Online Learning: A Comprehensive Survey*, review online learning algorithms that update a model one instance at a time, together with their theory and applications. The survey supports the idea of learning from a stream with constant memory, but as a survey it does not provide a drift-aware forecasting system, and the practical question of detecting drift and adapting quickly in forecasting is left open.

IV. THE BUFFER-FREE DRIFT-AWARE FORECASTING FRAMEWORK

Building on the reviewed work, we outline a conceptual Buffer-Free Drift-Aware Forecasting framework. It is a design proposal derived from this survey. It has not been implemented or evaluated here, and no performance claims are made for it. "Buffer-free" means that no sliding window, replay buffer or ensemble of models is kept: only a small set of running

statistics and the parameters of one model are stored. The framework has seven parts.

1. Stream input. The stream arrives one observation at a time, together with optional covariates. No past samples are stored. The only state kept between steps is the running statistics, the model parameters and the detector state.
2. Online normalisation. The running mean and variance are updated recursively with a forgetting factor, so recent values count more than old ones. The input is normalised with these statistics and the forecast is converted back to the original scale, in the spirit of [6] and [7] but without any window.
3. Incremental forecaster. One lightweight model, for example a small linear or neural forecaster, predicts the next value or the next few values. A single model is used on purpose, because the aim is to avoid the memory and latency of ensembles [5], [9] and memory modules [8].
4. Error monitoring. When the true value arrives, the absolute error is computed and the running mean and deviation of the error are updated recursively. Because the true value is known only after the horizon, the framework treats the error signal as delayed [11].
5. Drift detection. A sequential test on the error statistics, such as the Page-Hinkley test or an exponentially weighted moving average chart,

accumulates evidence at every step and signals drift when the accumulated deviation crosses a threshold. This follows the error-based idea of DDM [4] with a constant-size state.

6. Buffer-free adaptation. On a drift signal, the learning rate is raised and older information is forgotten faster, and the model is updated with one gradient step on the current observation [10]. When the error returns to normal, the learning rate is lowered again to keep the model stable.
7. Evaluation protocol. We recommend prequential evaluation on benchmark and synthetic streams with known drift points, comparison with buffer-based and ensemble-based baselines, and reporting of accuracy, adaptation delay, memory and latency as listed in Table II.

V. PERFORMANCE METRICS

Table I compares the reviewed studies. The descriptions are those given by the authors and are not directly comparable, because tasks, datasets, drift types and evaluation protocols differ. Entries marked “Not reported here” were not stated in the abstracts consulted, and the memory column is our qualitative reading of each method’s design, not a measurement. Table II lists the metrics and ablations we propose for evaluating the framework in Section IV.

Table I. Comparison of Reviewed Drift-Handling and Online Forecasting Studies

Study	Task	Drift handling	Memory demand (our reading)	Reported result / limitation
Gama et al. [1]	Stream learning (review)	Taxonomy of detection and adaptation	Mostly windows	Review; no forecasting method
Lu et al. [2]	Drift learning (review)	Detect, understand, adapt	Windows or ensembles	Framework only; no algorithm
Bifet and Gavalda [3] (ADWIN)	Change detection in streams	Adaptive window, mean test	Window stored; grows when stable	Guarantees on false positives and negatives; not designed for forecasting
Gama et al. [4] (DDM)	Classification streams	Error-rate warning and drift levels	Low for detection	Slow on gradual drift; rebuilds from samples
Gomes et al. [5] (ARF)	Stream classification	Per-tree detectors, background trees	High (many trees)	Heavy computation; classification only
Kim et al. [6] (RevIN)	Time series forecasting	Instance normalisation	Low	Input statistics only; no model update
Fan et al. [7] (Dish-TS)	Time series forecasting	Learned normalisation coefficients	Medium (offline, windows)	Offline training; extra networks
Pham et al. [8] (FSNet)	Online forecasting	Fast adapter and associative memory	Medium to high (memory module)	Efficacy on real and synthetic data; added complexity

Zhang et al. [9] (OneNet)	Online forecasting	Online ensemble of two models	High (two models)	Error reduced by more than 50% vs previous best; higher memory and latency
Hoi et al. [10]	Online learning (survey)	Sequential updates	Constant in principle	Survey; no drift-aware forecaster
Zhao and Shen [11] (Proceed)	Online forecasting	Drift estimate and adaptation generator	Medium (recent samples, generator)	Gains over earlier online learners on five datasets; pre-trained generator

Table II. Proposed evaluation protocol for the buffer-free forecasting framework

Aspect	What to measure	Variants / setting	Reason
Forecast accuracy	MAE and RMSE under prequential evaluation; error before, during and after drift	Report each dataset and each horizon separately	A single average hides the loss of accuracy right after a drift
Adaptation delay	Steps from a known drift point until the error returns to its pre-drift level	Synthetic streams with sudden, gradual, incremental and recurring drift	Shows how fast the method recovers
Memory	Peak memory and number of stored values	Compare with window, buffer and ensemble baselines	Tests the buffer-free claim directly
Latency	Processing time per observation, including detection and update	Measured on an edge-class CPU	Shows suitability for real-time use
Detector quality	False alarm rate and missed drift rate	Vary the detector threshold	Balances early detection against false alarms
Component ablation	Without normalisation; without detector; without drift-mode learning rate	Same forecaster and data	Isolates the contribution of each part
Baselines	Static model, periodic retraining, ADWIN or DDM triggered retraining, FSNet, OneNet	Same data and horizon	Shows the accuracy and cost trade-off against buffer-based methods [3], [4], [8], [9]

VI. ANALYSIS

Several patterns emerge from the reviewed work.

Drift handling is bought with memory. The methods that adapt best keep something extra: a window in ADWIN [3], many trees in ARF [5], an associative memory in FSNet [8] or two models in OneNet [9]. Memory and computation grow with the size of that extra component, which is a problem for edge devices. Normalisation is light but incomplete. RevIN [6] and Dish-TS [7] are cheap, but they act on the input statistics only. They help with virtual drift, but when the relation between input and target changes, the model itself has to change, and these methods do not update it.

Detectors and forecasters are rarely joined without a buffer. DDM [4] and ADWIN [3] were designed for classification streams or for the error signal, and the online forecasters [8], [9], [11] adapt the network but work from recently stored samples. In the abstracts we

consulted, we found no method that detects drift from the forecast error and adapts one model from the current observation alone.

Window size is a hidden parameter. A short window gives noisy detection and a long window reacts late [1], [3]. A buffer-free design removes this choice, but replaces it with settings such as the forgetting factor and the detector threshold, which also need tuning.

Evaluation seldom reports cost. The abstracts we consulted report accuracy or error, for example the reduction reported in [9], but not peak memory, latency or detection delay. Without these figures the benefit of a lighter method cannot be shown.

Open challenges. (i) A forecasting method that detects drift and adapts without windows, buffers or ensembles. (ii) Handling of gradual and incremental drift, where each step is small. (iii) Reporting of memory, latency and adaptation delay alongside error. (iv) Handling of delayed feedback, because the true value arrives only after the horizon [11]. (v) Stable

behaviour when drift is frequent or when false alarms occur. (vi) Realistic streams with labelled drift points. Future directions. Promising directions include multi-step forecasting with the framework, detector thresholds that tune themselves, drift-type-aware adaptation that treats sudden, gradual and recurring drift differently, a small memory of recurring concepts that stays within a fixed budget, tests on edge hardware, and federated variants in which devices share only model updates.

VII. CONCLUSION

Concept drift is a main obstacle to accurate real-time forecasting, because the patterns learned from past data lose their validity as the data changes. The reviewed studies show that drift can be detected and handled, that normalisation reduces the effect of shifts in input statistics, and that deep online forecasters can adapt quickly. At the same time, most of them rely on windows, buffers, memory modules or several models, which increases memory use, delay and privacy risk, and the evidence is limited by the lack of memory and latency reporting. A framework such as the proposed Buffer-Free Drift-Aware Forecasting design, which normalises the stream with running statistics, forecasts with one incremental model, detects drift from the prediction error and adapts from the current observation alone, is a reasonable candidate for constant memory and low latency. Whether it reaches the accuracy of buffer-based methods can only be settled by implementation and the evaluation protocol of Table II, which is the planned next step.

REFERENCES

- [1]. Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., and Bouchachia, A., “A survey on concept drift adaptation,” *ACM Computing Surveys*, vol. 46, no. 4, pp. 1–37, 2014.
- [2]. Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., and Zhang, G., “Learning under concept drift: A review,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 12, pp. 2346–2363, 2019.
- [3]. Bifet, A., and Gavaldà, R., “Learning from time-changing data with adaptive windowing,” in *Proceedings of the SIAM International Conference on Data Mining (SDM)*, 2007, pp. 443–448.
- [4]. Gama, J., Medas, P., Castillo, G., and Rodrigues, P., “Learning with drift detection,” in *Advances in Artificial Intelligence – SBIA 2004*, Lecture Notes in Computer Science, vol. 3171, Springer, 2004, pp. 286–295.
- [5]. Gomes, H. M., Bifet, A., Read, J., Barddal, J. P., Enembreck, F., Pfahringer, B., Holmes, G., and Abdesslem, T., “Adaptive random forests for evolving data stream classification,” *Machine Learning*, vol. 106, nos. 9–10, pp. 1469–1495, 2017.
- [6]. Kim, T., Kim, J., Tae, Y., Park, C., Choi, J.-H., and Choo, J., “Reversible instance normalization for accurate time-series forecasting against distribution shift,” in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.
- [7]. Fan, W., Wang, P., Wang, D., Wang, D., Zhou, Y., and Fu, Y., “Dish-TS: A general paradigm for alleviating distribution shift in time series forecasting,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.
- [8]. Pham, Q., Liu, C., Sahoo, D., and Hoi, S. C. H., “Learning fast and slow for online time series forecasting,” in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.
- [9]. Zhang, Y.-F., Wen, Q., Wang, X., Chen, W., Sun, L., Zhang, Z., Wang, L., Jin, R., and Tan, T., “OneNet: Enhancing time series forecasting models under concept drift by online ensembling,” in *Advances in Neural Information Processing Systems 36 (NeurIPS)*, 2023.
- [10]. Hoi, S. C. H., Sahoo, D., Lu, J., and Zhao, P., “Online learning: A comprehensive survey,” *Neurocomputing*, vol. 459, pp. 249–289, 2021.
- [11]. Zhao, L., and Shen, Y., “Proactive model adaptation against concept drift for online time series forecasting,” *arXiv preprint arXiv:2412.08435*, 2024.