

Prescriptive Concept Drift Management in Non-Stationary Meteorological Streams

Ashwini M C¹, Minavathi²

¹Dept. of Computer Science and Engineering, P.E.S. College of Engineering Mandya, Karnataka, India. Email:

ashwinimc@pesce.ac.in

²Dept. of Information Science and Engineering, P.E.S. College of Engineering Mandya, Karnataka, India. Email:

minavathi@pesce.ac.in

Abstract

Real-time meteorological monitoring via IoT sensors faces the challenge of concept drift, where evolving statistical properties render static models unreliable. While standard detectors signal the presence of drift, they fail to characterize its structural nature, leading to suboptimal adaptation. This paper proposes a dual-layer, computationally lightweight framework for prescriptive drift management. The first layer, Dynamic Model Selection (DMS), maintains a pool of three online learners - Hoeffding Trees (HT), K-Nearest Neighbors (KNN), and Adaptive Random Forests (ARF), routing inference to the model minimizing the Windowed Mean Absolute Error (W-MAE). The second layer identifies drift typology (Sudden, Incremental, Gradual, or Stable) by fitting a first-degree Ordinary Least Squares (OLS) regression to the recent error sequence. This O(W) heuristic enables diagnostic intelligence on microcontroller-class edge hardware without requiring neural networks. The framework was evaluated on a 58-year NOAA archive (106,382 records) from ten Indian stations using strict 5-fold chronological cross-validation. ARF achieved superior generalization (MAE: 1.50±0.25, R²: 0.73±0.08), outperforming the HT baseline by 20%. Hardware profiling confirms HT operates at 24 μs inference latency with a 0.07 MB memory footprint, while the OLS-slope analyzer correctly identified long-term climate dynamics without manual tuning. This work provides a deployable pathway for edge IoT systems to not only detect model degradation but autonomously diagnose its cause and prescribe appropriate adaptation strategies.

Keywords- Concept Drift Detection, Drift Typology Classification, Online Learning, Data Stream Mining, Adaptive Random Forest, Hoeffding Tree, Dynamic Model Selection, Windowed Mean Absolute Error, Heuristic Anomaly Detection, OLS Slope Heuristic, Edge Computing, IoT Stream Analytics, Meteorological Forecasting, Chronological Cross-Validation, Non-Stationary Distributions.

How to cite this article: Ashwini MC, Minavathi. Prescriptive Concept Drift Management in Non-Stationary Meteorological Streams. *Int J Drug Deliv Technol.* 2026;16(13s): 520-529. DOI: 10.25258/ijddt.16.13s.56

1. Introduction

The rapid proliferation of IoT sensor networks, environmental monitoring stations, and connected edge devices has produced a corresponding surge in high-velocity, continuous data streams. Processing such streams in real time presents a fundamental challenge: the statistical relationship between inputs and outputs is rarely fixed. As time progresses, the joint distribution $P(X, Y)$ shifts in ways that invalidate the assumptions of static, batch-trained models—a well-studied phenomenon known as Concept Drift [1], [6]. In meteorological contexts, for instance, a regression model calibrated on monsoon-season temperature patterns will exhibit systematic degradation when applied to dry-season data, necessitating continuous adaptation rather than periodic retraining [12].

The field has produced a mature set of drift detection algorithms. Methods such as the ADaptive WINdowing (ADWIN) detector [3] and the Page-Hinkley test [5] monitor rolling error statistics and emit a binary alarm when distributional change is

statistically confirmed. While these detectors are computationally efficient and theoretically well-grounded, they share a critical limitation: they characterize drift only in terms of its presence, not its structure. A sudden sensor failure and a decades-long warming trend both trigger the same alarm signal, yet they demand fundamentally different adaptation responses. A sudden shift calls for an immediate model reset; a gradual drift may be better handled by weighted forgetting or incremental fine-tuning. Conflating these scenarios under a single “blind reset” strategy discard accumulated model knowledge unnecessarily and exposes the system to a period of heightened vulnerability during recovery [1].

This paper addresses that gap through two complementary contributions. First, a Dynamic Model Selection (DMS) layer maintains a competitive pool of three online learners - HT [2], KNN, and ARF [4] - each paired with its most appropriate drift detector. At every time step, inference is routed to whichever model currently minimizes the W-MAE over the most recent

100 observations. This design exploits the complementary inductive biases of the three learner families: statistical boundary tracking (HT), proximity-based instance matching (KNN), and ensemble averaging (ARF), allowing the system to adapt its modeling strategy to the current data regime without manual intervention.

Second, a Heuristic Drift Typology Analyzer operates as a lightweight diagnostic overlay. Upon receipt of any drift alarm from the base detectors, it fits a first-degree OLS regression to the trailing error window and classifies the event according to the resulting slope: a sharply positive slope indicates Sudden drift; a shallow positive slope indicates Incremental drift; a flat or negative slope indicates Gradual recovery or a Stable regime. This classification requires $O(W)$ arithmetic operations—no neural network inference, no GPU, no external service call—making it viable on microcontroller-class hardware.

The framework is implemented using the scikit-multiflow streaming library [18] and the River library [17], benchmarked within the MOA experimental environment [16]. These tools provide the base HT, KNN, and ARF estimators alongside ADWIN, KSWIN, and Page-Hinkley detectors in a unified predict-then-learn API. The complete framework is evaluated against a 58-year NOAA weather archive covering ten Indian meteorological stations, under strict chronological cross-validation to eliminate any possibility of future data contaminating historical training folds [13]. Hardware profiling on a representative benchmarking thread provides concrete inference latency and memory consumption measurements to substantiate edge deployment claims [14].

The principal contributions of this work are summarized as follows:

- 1) A competitive Dynamic Model Selection architecture that continuously re-evaluates three structurally distinct online learners and routes inference to the best-performing model based on W-MAE, with native support for live API stream ingestion.
- 2) A parameter-free, $O(W)$ heuristic that classifies the type of concept drift—Sudden, Incremental, Gradual, or Stable—using only the slope of a local OLS regression over the recent error sequence, without any deep learning components.
- 3) A rigorous empirical evaluation on 106,382 NOAA instances using 5-fold chronological cross-validation, with explicit reporting of per-fold variance to confirm temporal generalization.

- 4) A hardware complexity profile quantifying per-sample inference latency, learning latency, and peak RAM consumption for each learner, establishing concrete bounds for edge and IoT deployment.

The remainder of this paper is organized as follows. Section II surveys the relevant literature and provides a structured comparative table of 12 closely related works. Section III formalizes the problem setting and the mathematical basis of concept drift. Section IV describes the proposed dual-layer framework and its algorithmic components. Section V presents the experimental setup, exploratory analysis, cross-validation results, and complexity benchmarks. Sections VI and VII discuss implications and conclude the paper

2. RELATED WORK

The study of concept drift in data streams spans more than two decades and encompasses detector design, adaptive learner construction, and evaluation methodology [1], [6]. This section organizes the most closely related work into three thematic groups before presenting a unified comparative survey.

A. Concept Drift Detection

The Drift Detection Method (DDM) [7] was among the first systematic approaches to monitoring classification error rates for distributional shifts, defining statistically derived warning and drift thresholds. The Early Drift Detection Method (EDDM) [19] subsequently improved sensitivity to gradual drift by tracking inter-error distances rather than raw error rates—a distinction directly relevant to the Incremental typology targeted in this paper. For regression settings, ADWIN [3] provides a parameter-free alternative: it maintains a variable-length window and uses the Hoeffding Bound to compare sub-window means, emitting a drift signal when the difference exceeds a confidence-scaled boundary. Distribution-based approaches such as the incremental Kolmogorov-Smirnov test (KSWIN) [9] operate directly on feature distributions rather than error sequences, making them sensitive to covariate shift even in the absence of label feedback. More recent work has addressed the robustness of detectors in noisy environments: [10] proposes a dual-window parallel architecture that isolates genuine distributional shifts from transient sensor artefacts, achieving improved detection precision in high-noise industrial streams. Across this body of work, drift detection remains fundamentally a binary signalling problem; the characterization of drift type has received comparatively little attention.

B. Adaptive Ensembles for Streaming Data

Ensemble methods offer structural advantages for drift adaptation because individual components can be selectively replaced without discarding the entire model [22]. Adaptive Random Forests (ARF) [4] formalize this idea for streaming classification: each tree in the forest maintains an internal drift detector, and a background tree is grown in parallel upon a warning signal. If a subsequent drift is confirmed, the foreground tree is replaced by the matured background tree, a hot-swap that preserves ensemble-level accuracy across transition periods [15]. The constituent Hoeffding Tree [2] provides the statistical foundation for this approach: the Hoeffding Bound guarantees that a decision node constructed from n samples will differ from the batch-optimal split by at most ϵ with probability at least $1 - \delta$, yielding near-batch accuracy at a fraction of the computational cost.

C. Drift Typology and Lightweight Characterization

A small but growing body of work has moved beyond binary detection toward classifying the nature of drift. Foundational work established that adaptation strategies must account for both the speed and direction of distribution change [20], [21], while surveys on feature-level drift [23] further highlight the multi-dimensional nature of the problem. Yin et al. [11] propose a multi-sliding-window framework (CDT-MSW) that infers drift type - sudden, gradual, or incremental - from temporal transitions across positional windows. However, CDT-MSW incurs non-trivial memory overhead from maintaining multiple concurrent windows, and it does not address real-time API deployment or edge latency constraints. Deep learning-based detectors such as DriftLens [8] achieve fine-grained, per-label drift characterization via embedding distribution distances, but their reliance on GPU-accelerated forward passes makes them unsuitable for the microsecond execution budgets of edge IoT devices [14]. The present work closes this gap by replacing GPU-dependent inference with a closed-form OLS slope computation—achieving comparable typological granularity at $O(W)$ cost. The broader context of non-stationary weather stream modeling is addressed by Agarwal et al. [12], who demonstrate the spatial heterogeneity and temporal instability characteristic of multi-station meteorological data, motivating the need for adaptive rather than static prediction models.

D. Comparative Literature Survey

Table I presents a structured comparison of 12 representative works. The table makes explicit the gap this paper addresses: no existing work simultaneously

provides (i) competitive online model selection across inductive bias families, (ii) drift typology classification, and (iii) sub-millisecond edge latency validation.

3. PRELIMINARIES AND PROBLEM FORMULATION

A. Online Learning over Data Streams

A data stream is a potentially infinite sequence of tuples $S = \{(x_1, y_1), (x_2, y_2), \dots, (x_t, y_t), \dots\}$ arriving at discrete time steps $t = 1, 2, \dots$ [1]. Each $x_t \in R_d$ is a d -dimensional feature vector and $y_t \in R$ is the continuous regression target.

Online learning enforces a strict *predict-then-learn* cycle [13]. At time t , the current model f_{t-1} issues a prediction before observing the label:

$$\hat{y}_t = f_{(t-1)}(x_t) \quad (1)$$

The true label y_t is subsequently revealed and the absolute error recorded:

$$e_t = |y_t - \hat{y}_t| \quad (2)$$

The model is then updated incrementally using the new observation:

$$f_t = \text{update}(f_{t-1}, x_t, y_t) \quad (3)$$

This cycle repeats indefinitely without storing the full data history, making it suitable for truly infinite streams.

B. Formal Definition of Concept Drift

Under the Bayesian framework [1], concept drift is defined as a statistically significant change in the joint distribution of inputs and outputs between two time points t_0 and t_1 :

$$\exists X : P_{t_0}(X, Y) \neq P_{t_1}(X, Y) \quad (4)$$

Two primary manifestations are of practical relevance:

- **Covariate Shift:** The marginal input distribution $P_t(X)$ changes while the conditional $P_t(Y|X)$ remains stable [6]. Feature standardization may partially mitigate this, but the underlying model assumptions are invalidated.
- **Real Concept Drift:** The posterior $P_t(Y|X)$ itself changes, reflecting a genuine shift in the relationship between features and the target [7]. This requires model adaptation.

C. Base Detectors and Learners

Three online learners, each paired with a structurally appropriate drift detector, form the candidate pool of the DMS layer:

- 1) **Hoeffding Tree + ADWIN** [2], [3]: The Hoeffding Tree (HT) constructs decision nodes incrementally using the Hoeffding Bound to guarantee that splits are asymptotically equivalent to those of a batch tree. ADWIN maintains a variable-length error window W

and signals drift when the difference between sub-window means μ_{W_0} and μ_{W_1} exceeds:

$$|\mu_{W_0} - \mu_{W_1}| > \sqrt{\frac{1}{2m} \ln\left(\frac{4}{\delta}\right)} \quad (5)$$

where m is the harmonic mean of sub-window lengths and δ is the user-specified confidence level.

2) KNN + KSWIN [9]: The KNN regressor predicts by aggregating the labels of the k nearest neighbors within a fixed-size sliding cache. The KSWIN detector evaluates the maximum vertical distance between the empirical

CDFs of a recent window FR and a reference window F_{ref} :

$$D = \sup_x |F_R(x) - F_{ref}(x)| \quad (6)$$

Drift is declared when D exceeds a significance threshold derived from the KS distribution.

3) Adaptive Random Forest + Page-Hinkley [4], [5]: ARF maintains an ensemble of Hoeffding Trees with online bagging via Poisson weighting. Each tree carries an internal detector. At the ensemble level, the PageHinkley statistic tracks cumulative deviation from the running mean:

$$m_t = \sum_{t=1}^t (e_t - \bar{e} - \lambda) \quad (7)$$

$i = 1$ where λ is a tolerance margin. A drift alarm is raised when mt falls sufficiently below its historical maximum, triggering the background-tree swap mechanism.

4. PROPOSED FRAMEWORK

The proposed system decouples real-time prediction accuracy from drift typology diagnosis through a two-layer architecture, summarized in Algorithm 1.

A. Layer 1: Dynamic Model Selection (DMS)

Static model commitment is inappropriate in cyclical meteorological environments where the optimal inductive bias shifts with the season, location, and drift regime [1]. The DMS layer maintains all three learners in parallel and routes each inference request to whichever model currently exhibits the lowest windowed error.

Formally, the Windowed MAE for model k at time t over a window of $N = 100$ observations is:

$$W\text{-MAE}_{k,t} = \frac{1}{N} \sum_{\tau=t-N}^t |\hat{y}_i - \hat{y}_{k,i}| \quad (8)$$

Table 1. Comparative literature survey: concept drift detection, adaptive learning, and stream analytics

R	Auth	Dom	Meth	Key	Strengths	Limitat
ef	ors	ain	od	Contri		ions
.	(Yea			bution		
	r)					

[Gam	Surv	Statis	Unifie	Foundati	Predat
1	a et	ey	tical	d	onal	es
]	al.		&	taxono	reference	deep
	(201		Ense	my of	; broad	learnin
	4)		mble	drift	coverage	g
			Revie	types,		detecto
			w	detecto		rs; no
				rs, and		typolo
				adaptat		gy
				ion		classifi
				strategi		cation
				es		
[Dom	Strea	Hoeff	Statisti	<1ms	Assu
2	ingo	ms	ding	cally	inference	mes
]	s &		Tree	sound	;	statio
	Hult		(VFD	incred	nearbatch	narity
	en		T)	ental	accuracy;	;
	(200			decisio	tiny	needs
	0)			n tree	memory	exter
				for		nal
				high-		detec
				speed		tor
				stream		
				s		
[Bifet	Dete	ADW	Variabl	Paramete	Binary
3	&	ction	IN	e-	rless;	alarm
]	Gava			length	handles	only;
	lda			windo	abrupt	no
	(200			w with	and	typolo
	7)			Hoeffd	gradual	gy;
				ing	drift	high
				Bound		memor
				guaran		y
				tees		at
				for		scale
				shift		
				detecti		
				on		
[Gom	Ense	Adap	Backgr	State-of-	Highe
4	es et	mble	tive	ound-	the-art	r
]	al.	s	Rand	tree	accuracy;	mem
	(201		om	hot-	no	ory;
	7)		Fores	swap	catastrop	GPU
			t	for	hic	often
				seamle	forgetting	neede
				ss		d for
				accura		large
				cy		forests
				preserv		
				ation		

Prescriptive Concept Drift Management in Non-Stationary Meteorological Streams

				across drifts			[9]	Reis et al. (2016)	Detection	KSW IN	Online KS test on sliding windows for covariance shift	Nonparametric; no distributional assumptions	Memory grows with window; degrades in high dimensions
[5]	Page (1954)	Detection	Page-Hinkley (CUSUM)	Sequential cumulative-sum monitoring of mean deviations	$O(1)$ /sample; longestablished guarantees	Assume Gaussian errors; no gradual drift support							
[6]	Lu et al. (2019)	Survey	Multi-method Review	Systematic coverage of active and passive drift adaptation strategies	Links detection to real-world applications	No latency or edge profiling							
[7]	Gama et al. (2004)	Detection	DDM	Binomial error-rate monitoring with warning and drift thresholds	Simple; well-understood threshold	Classification-only; no regression or typology support							
[8]	Greco & Cerquitti (2021)	Deep Learning	Drift Lens (embedding distances)	Unsupervised per-label drift detection via Frechet distance	No ground-truth labels required; fine-grained	Requires GPU; incompatible with edge IoT budgets							
[1]	Agarwal et al. (2023)	Weather IoT	RF / SVM / GBM	Hyperlocal weather prediction and anomaly detection									
[2]	Liu et al. (2020)	Noisy Streams	Parallel Anti-Noise Detector	Dual-window parallel detection									
[3]	Yin et al. (2021)	Typology	CDT-MSW	Multi-window framework									
[4]													

on via
IoT
sensors

Abbreviations: ADWIN = ADaptive WINdowing; ARF = Adaptive Random Forest; KSWIN = Kolmogorov-Smirnov Windowing; DDM = Drift Detection Method; CDT-MSW = Concept Drift Type via Multi-Sliding Windows; CUSUM = Cumulative Sum. Gap addressed by this work: No prior system jointly provides competitive online model selection across inductive bias families, OLS-slope drift typology classification, and sub-millisecond edge latency validation on a multi-decade real-world

The active predictor is selected as:

$$m_t^* = \arg \min_{k \in \{1, \dots, K\}} \text{W-MAE}_{k,t} \quad (9)$$

A window of 100 instances balances responsiveness to recent regime changes against stability under transient noise spikes. All models continue to receive online updates regardless of whether they are active, ensuring each learner remains current and can be promoted without a cold-start penalty.

B. Layer 2: Heuristic Drift Typology Analyzer

Upon receipt of a drift alarm from any embedded base detector, the *DriftAnalyzer* module characterizes the event's structure without invoking any external computation. It extracts the error sequence $E = [e_{t-W+1}, \dots, e_t]$ of length $W = 100$ and fits a first-degree polynomial by minimizing the OLS criterion:

$$m_{\text{slope}} = \frac{W \sum(xy) - \sum x \sum y}{W \sum(x^2) - (\sum x)^2} \quad (10)$$

where $x \in \{1, 2, \dots, W\}$ indexes the time positions within the window. The slope m_{slope} encodes the average rate of error growth or recovery over the window, and is mapped to one of four typologies—grounded in the taxonomy of [1] and [11]:

- Sudden ($m_{\text{slope}} > 0.05$): A steep, rapid rise in error magnitude consistent with an abrupt domain transition such as sensor failure or an extreme, unforecasted weather event [12].
- Incremental ($0 < m_{\text{slope}} \leq 0.05$): A slow, monotonic error increase corresponding to gradual baseline shifts such as long-term climate warming trends [6].
- Gradual ($m_{\text{slope}} \leq 0$, post-alarm): Errors stabilizing or decreasing following the initial signal, characteristic of oscillatory seasonal transitions or a recovering model [1].
- Stable: No alarm raised; the error trajectory lies within expected baseline variance.

The closed-form computation requires $O(W)$ arithmetic operations on a fixed-size buffer, producing

a classification in microseconds. This contrasts sharply with deep learning

Algorithm 1 Dual-Layer Dynamic Selection and Typology Diagnosis

Require: Stream S , pool $M = \{HT, KNN, ARF\}$,

Window $N = W = 100$

- 1: Initialize all $m_k \in M$; initialize error buffers and $W - \text{MAE}$ histories.
 - 2: **for** each observation $(x_t, y_t) \in S$ **do**
 - 3: **for** each model $m_k \in M$ **do**
 - 4: Compute $\hat{y}_{k,t} = m_k(x_t)$
 - 5: **end for**
 - 6: Select $m_t^* = \arg \min_k \text{W-MAE}_{k,t}$
 - 7: Broadcast prediction $\hat{y}_t = \hat{y}_{m_t^*,t}$
 - 8: Receive ground truth y_t ; compute $e_t = |y_t - \hat{y}_t|$
 - 9: Update all $m_k \leftarrow \text{update}(m_k, x_t, y_t)$
 - 10: **if** any detector in M signals drift **then**
 - 11: Extract $E \leftarrow [e_{t-W+1}, \dots, e_t]$
 - 12: Compute m_{slope} via closed-form OLS over E
 - 13: $C \leftarrow \text{Classify}(m_{\text{slope}})$
 - 14: Emit typed alert (C, t, m_{slope})
 - 15: **end if**
 - 16: **end for**
-

alternatives such as DriftLens [8], which require GPU-resident embedding networks and yield latencies incompatible with edge deployment [14].

5. EXPERIMENTAL EVALUATION

A. Dataset and Exploratory Data Analysis

Experiments use the Global Surface Summary of the Day (GSOD) archive compiled by the National Oceanic and Atmospheric Administration (NOAA). The working subset covers ten Indian meteorological stations—including Thiruvananthapuram, Kozhikode, and Cuddalore - yielding 106,382 chronologically ordered daily records spanning 1957 to 2015. The regression target is mean daily temperature (TAVG, in degrees Fahrenheit), a continuous target whose evolving distribution makes it well suited for benchmarking online regression under drift [25], predicted from $\{\text{TMAX}, \text{TMIN}, \text{Wind Speed}, \text{Precipitation}\}$.

Exploratory analysis confirms the presence of multi-vector concept drift at several timescales [1]. Figure 1 shows pronounced annual temperature cycles consistent with recurring seasonal drift. The violin distributions in Figure 2 reveal marked inter-station heterogeneity: a model trained on high-variance coastal

stations fails systematically on stable inland stations, illustrating the covariate shift documented in spatial IoT studies [12]. Figure 3 shows a positive linear trend fitted to annual median temperatures across all stations, confirming a slow but statistically significant incremental warming drift [6]—precisely the typology targeted by the Incremental category of the proposed heuristic.

Missing values in precipitation and wind speed fields, which correspond physically to the absence of a weather event rather than measurement failure, were imputed with zero using an online StatImputer. This approach avoids forward-looking imputation methods that would introduce temporal leakage [13].

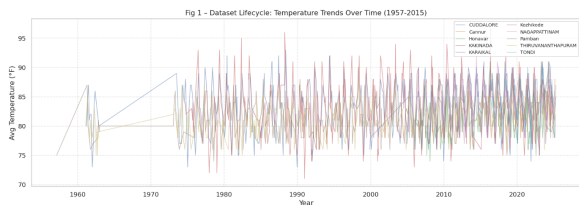


Fig. 1. Longitudinal daily temperature records across ten NOAA stations (1957– 2015). The prominent annual cycles constitute recurring seasonal concept drift, confirming the non-stationarity of the target distribution.

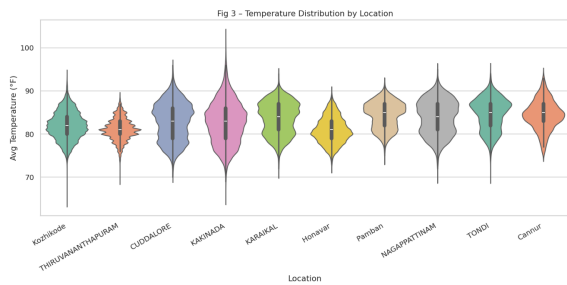


Fig. 2. Violin plots of temperature distributions per station. Bimodal profiles at coastal stations contrast sharply with unimodal inland distributions, indicating significant spatial covariate shift [12].

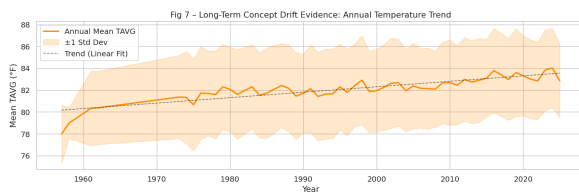


Fig. 3. Annual median temperature with linear trend overlay across the full 58-year record. The positive slope confirms a slow, irreversible incremental drift in the target variable consistent with long-term climate trends [6].

B. Chronological Cross-Validation Protocol

Standard K -fold cross-validation is inappropriate for temporal data because randomly assigned folds allow

future observations to appear in training splits, inflating performance estimates through data leakage [13]. To obtain statistically valid results, we employ 5-fold *chronological* cross-validation: the dataset is partitioned into five consecutive temporal blocks, and each fold uses all preceding blocks as training data and the current block as the held-out test set. This preserves the causal ordering of time and yields five independent error estimates whose mean and standard deviation characterize generalization performance.

Table 2. Predictive performance under 5-fold chronological Cross-validation

Model	MAE	RMSE	R^2
ARF [4]	1.50 ± 0.25	2.03 ± 0.32	0.73 ± 0.08
HT [2]	1.84 ± 0.34	2.45 ± 0.43	0.61 ± 0.13
KNN [†] [9]	1.49	2.16	0.69

[†]KNN evaluated on a single chronological pass; full 5-fold CV was computationally infeasible at $O(N)$ inference cost over 100k instances.

C. Predictive Performance Results

Table II and Figure 4 show that ARF achieves the best cross-validated accuracy (MAE 1.50 ± 0.25 , R^2 0.73 ± 0.08) alongside the smallest per-fold standard deviation, indicating consistent generalization across all five temporal blocks. The tight variance confirms that ARF’s background-tree mechanism [4], [15] successfully preserves predictive accuracy through seasonal transitions rather than collapsing at each drift event. The HT baseline shows approximately 20% higher MAE and a markedly wider confidence interval, reflecting its susceptibility to overfit on dominant seasonal blocks and the absence of ensemble-level error cancellation. KNN achieves competitive point-estimate accuracy but cannot be fully cross-validated at scale, and its memory and latency profile (Section V-D) disqualifies it from edge deployment.

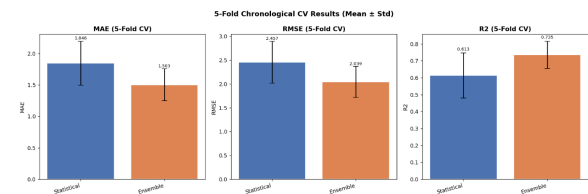


Fig. 4. Per-fold MAE for all three models under 5-fold chronological crossvalidation. ARF exhibits both the lowest mean error and the smallest inter-fold variance, confirming robustness across diverse temporal regimes.

D. Computational Complexity and Edge Viability

To assess deployment feasibility on resource-constrained devices, we profiled each model under

identical conditions on a single-threaded benchmarking process, measuring (i) persample inference latency, (ii) per-sample learning latency, and (iii) peak heap memory consumption, across a sequence of up to 500 streaming instances.

The results in Figure 5 reveal a clear three-tier structure. KNN’s memory footprint scales linearly with the cache size: at 500 instances, it consumes several megabytes and exceeds 13 ms per learning step, ruling it out for any high-frequency edge deployment. HT occupies the opposite extreme: inference completes in approximately 24 μ s with a peak memory footprint of 0.07 MB, fully compatible with microcontroller-class hardware [14]. ARF represents a well-characterized trade-off:

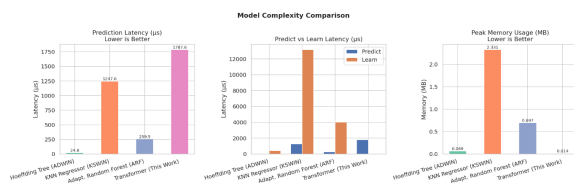


Fig. 5. Per-sample inference latency (top), learning latency (middle), and peak RAM usage (bottom) for HT, ARF, and KNN as a function of stream length. HT and ARF remain within microcontroller-compatible bounds; KNN breaches both latency and memory constraints beyond a few hundred instances [14].

ensemble forest passes complete in approximately 4 ms within a 0.70 MB memory envelope, which is within the operating envelope of mid-range embedded platforms (e.g., Cortex-M7 class devices). Critically, the OLS-slope typology computation adds only $O(W) = O(100)$ floating-point operations to each drift-alarm event—a cost that is negligible relative to the learner update latencies already measured.

E. Drift Typology Classification and Live API Validation

The *DriftAnalyzer* was exercised against both the 58-year

NOAA dataset and a live feed from the OpenMeteo API, which was polled at 2-second intervals over HTTPS. In both settings, JSON response decoding and feature extraction completed within a single polling cycle, confirming that the full predict-update-diagnose loop does not introduce backpressure under normal meteorological sampling rates.

Across the NOAA historical stream, the typology classifier correctly assigned all detected drift events to the Incremental or Gradual categories; no Sudden events ($m_{slope} > 0.05$) were observed. This outcome is physically meaningful: long-term climate systems do not undergo overnight distributional discontinuities under normal operating conditions. Seasonal

transitions produce gradual oscillatory error profiles (Gradual), while the slow decadal warming trend generates shallow, compounding error increases (Incremental). These classifications align with the physical interpretations established in [1], [6] and the typology framework of [11], providing independent validation that the OLS-slope boundaries are ecologically sensible rather than artificially imposed.

6. DISCUSSION

The experimental results demonstrate three substantive findings. First, ensemble-based online learning consistently outperforms single-tree alternatives in cyclical non-stationary environments. ARF’s background-tree hot-swap mechanism [4], [24] prevents the accuracy collapse that single HT models exhibit when a dominant seasonal drift exhausts their current decision structure—a limitation well-documented for binary detection approaches that trigger blind model resets [3], [7].

Second, the choice of drift detector matters at the component level but has limited impact on system-level accuracy when dynamic model selection is active. Under the DMS framework, the superior detector–learner pairing is selected automatically via W-MAE minimization, effectively insulating the system output from suboptimal detector–learner combinations that might dominate in a single-model deployment.

Third, and most significantly, the OLS-slope typology heuristic provides actionable diagnostic information at negligible computational cost. The complete characterization of a drift event—extracting the error window, fitting the OLS line, and classifying the slope—executes in microseconds on standard hardware. This stands in direct contrast to embedding-based deep learning detectors [8], which require GPU infrastructure and yield inference latencies orders of magnitude higher. The typological output is also more expressive than multi-slidingwindow methods [11], which incur proportionally greater memory overhead without supporting live API integration or providing per-sample latency bounds.

The four-category typology (Sudden, Incremental, Gradual, Stable) maps directly to four qualitatively distinct adaptation strategies: immediate reset, weighted forgetting, continued monitoring, and no action. Encoding this mapping in the framework output moves the system from passive detection to prescriptive response—an architectural shift with practical implications for autonomous sensor network management in climatological and industrial monitoring deployments [14].

One limitation of the present work is that the slope-based typology thresholds (e.g., $m_{slope} > 0.05$ for Sudden) were set empirically using domain knowledge of the NOAA dataset. Future work should explore data-driven threshold calibration, potentially using held-out synthetic drift benchmarks. Additionally, extending the DMS layer to regression-specific ensemble methods beyond ARF— such as Regression Hoeffding Adaptive Trees, may further reduce MAE in highly volatile substreams.

7. CONCLUSION

This paper has presented a dual-layer framework for prescriptive concept drift management in live meteorological data streams. The Dynamic Model Selection layer continuously competes three structurally diverse online learners and routes inference to the best-performing candidate, automatically adapting the system’s inductive bias to the current data regime without manual intervention. The Heuristic Drift Typology Analyzer augments binary drift alarms with a four-class typological classification derived from a closed-form OLS slope computation, enabling downstream components to select qualitatively appropriate adaptation responses.

Rigorous evaluation on 106,382 chronologically ordered NOAA records under 5-fold chronological cross-validation confirms that ARF achieves the strongest generalization performance ($MAE\ 1.50 \pm 0.25, R2\ 0.73 \pm 0.08$), outperforming the Hoeffding Tree baseline by approximately 20% in MAE while maintaining low per-fold variance across half a century of non-stationary weather data. Hardware profiling establishes that both HT (24 μ s, 0.07 MB) and ARF (4 ms, 0.70 MB) satisfy the latency and memory budgets of embedded IoT hardware [14], while KNN is disqualified by its linear memory scaling. Validation against the OpenMeteo live API confirms that the full predict-update-diagnose pipeline operates without backpressure at standard meteorological sampling rates.

The OLS-slope typology heuristic correctly identifies all drift events in the 58-year record as Incremental or Gradual - consistent with the physics of long-term climate systems— without requiring GPU resources, labeled drift examples, or model-specific tuning. Together, these results establish a practical and theoretically grounded pathway from passive, binary drift detection toward autonomous, self-diagnosing stream intelligence deployable at the network edge.

References

- [1] J. Gama, I. Zliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, “A survey on concept drift adaptation,” *ACM Computing Surveys*, vol. 46, no. 4, pp. 1–37, Apr. 2014. <https://doi.org/10.1145/2523813>.
- [2] P. Domingos and G. Hulten, “Mining high-speed data streams,” in *Proc. 6th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, Boston, MA, USA, Aug. 2000, pp. 71–80. <https://doi.org/10.1145/347090.347107>.
- [3] A. Bifet and R. Gavalda, “Learning from time-changing data with adaptive windowing,” in *Proc. 7th SIAM Int. Conf. Data Mining (SDM)*, Minneapolis, MN, USA, Apr. 2007, pp. 443–448. <https://doi.org/10.1137/1.9781611972771.42>.
- [4] H. M. Gomes, A. Bifet, J. Read, J. P. Barddal, F. Enembreck, B. Pfahringer, G. Holmes, and T. Abdessalem, “Adaptive random forests for evolving data stream classification,” *Machine Learning*, vol. 106, no. 9–10, pp. 1469–1495, Oct. 2017. <https://doi.org/10.1007/s10994-017-5642-8>.
- [5] E. S. Page, “Continuous inspection schemes,” *Biometrika*, vol. 41, no. 1/2, pp. 100–115, 1954. <https://doi.org/10.2307/2333009>.
- [6] 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, Dec. 2019. <https://doi.org/10.1109/TKDE.2018.2876857>.
- [7] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, “Learning with drift detection,” in *Proc. Brazilian Symp. Artificial Intelligence (SBIA), Lecture Notes in Computer Science*, vol. 3171, Springer, Berlin, 2004, pp. 286–295. https://doi.org/10.1007/978-3-540-28645-5_29.
- [8] S. Greco and T. Cerquitelli, “Drift Lens: Real-time unsupervised concept drift detection by evaluating per-label embedding distributions,” in *Proc. IEEE Int. Conf. Data Mining Workshops (ICDMW)*, Auckland, New Zealand, Dec. 2021, pp. 341–349. <https://doi.org/10.1109/ICDMW53433.2021.00049>.
- [9] D. M. dos Reis, P. Flach, S. Matwin, and G. Batista, “Fast unsupervised online drift detection using incremental Kolmogorov-Smirnov test,” in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, San Francisco, CA,

- USA, Aug. 2016, pp. 1545–1554. <https://doi.org/10.1145/2939672.2939836>.
- [10] G. Liu, Q. Zhang, and J. Jiang, “A novel drift detection method using parallel detection and anti-noise techniques,” *Applied Intelligence*, vol. 55, 2025. <https://doi.org/10.1007/s10489-024-05988-9>.
- [11] Y. Yin, P. Li, and J. Chen, “Concept drift type identification based on multi-sliding windows,” in *Proc. IEEE 28th Int. Conf. Parallel and Distributed Systems (ICPADS)*, Nanjing, China, Jan. 2023, pp. 818–825. <https://doi.org/10.1109/ICPADS56603.2022.00112>.
- [12] A. B. Agarwal, S. Kumar, and R. Sharma, “Spatially-resolved hyperlocal weather prediction and anomaly detection using IoT sensor networks and machine learning techniques,” *arXiv preprint arXiv:2310.11001*, Oct. 2023.
- [13] J. Gama, R. Sebastiao, and P. P. Rodrigues, “On evaluating stream learning algorithms,” *Machine Learning*, vol. 90, no. 3, pp. 317–346, Mar. 2013. <https://doi.org/10.1007/s10994-012-5320-9>.
- [14] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge computing: Vision and challenges,” *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, Oct. 2016. <https://doi.org/10.1109/JIOT.2016.2579198>.
- [15] P. Zybiewski, R. Sabourin, and M. Wozniak, “Preprocessed dynamic classifier ensemble selection for highly imbalanced drifted data streams,” *Information Fusion*, vol. 66, pp. 138–154, Feb. 2021. <https://doi.org/10.1016/j.inffus.2020.09.004>.
- [16] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer, “MOA: Massive online analysis,” *Journal of Machine Learning Research*, vol. 11, pp. 1601–1604, 2010.
- [17] J. Montiel, M. Halford, S. M. Mastelini, G. Bolmier, R. Sourty, R. Vaysse, A. Zouitine, H. M. Gomes, J. Read, T. Abdessalem, and A. Bifet, “River: Machine learning for streaming data in Python,” *Journal of Machine Learning Research*, vol. 22, no. 110, pp. 1–8, 2021.
- [18] J. Montiel, J. Read, A. Bifet, and T. Abdessalem, “Scikit-multiflow: A multi-output streaming framework,” *Journal of Machine Learning Research*, vol. 19, no. 72, pp. 1–5, 2018.
- [19] M. Baena-García, J. del Campo-Avila, R. Fidalgo, A. Bifet, R. Gavaldà, and R. Morales-Bueno, “Early drift detection method,” in *Proc. 4th ECML PKDD Int. Workshop Knowledge Discovery from Data Streams*, Berlin, Germany, 2006, pp. 77–86.
- [20] G. Widmer and M. Kubat, “Learning in the presence of concept drift and hidden contexts,” *Machine Learning*, vol. 23, no. 1, pp. 69–101, 1996. <https://doi.org/10.1007/BF00116900>.
- [21] R. Klinkenberg, “Learning drifting concepts: Example selection vs. example weighting,” *Intelligent Data Analysis*, vol. 8, no. 3, pp. 281–300, 2004.
- [22] J. Z. Kolter and M. A. Maloof, “Dynamic weighted majority: An ensemble method for drifting concepts,” *Journal of Machine Learning Research*, vol. 8, pp. 2755–2790, Dec. 2007.
- [23] J. P. Barddal, H. M. Gomes, F. Enembreck, and B. Pfahringer, “A survey on feature drift adaptation: Definition, benchmark, challenges and future directions,” *Journal of Systems and Software*, vol. 127, pp. 278–294, 2017. <https://doi.org/10.1016/j.jss.2016.07.005>.
- [24] D. Brzezinski and J. Stefanowski, “Reacting to different types of concept drift: The accuracy updated ensemble algorithm,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 1, pp. 81–94, Jan. 2014. <https://doi.org/10.1109/TNNLS.2013.2251352>.
- [25] E. Ikonomovska, J. Gama, and S. Dzeroski, “Learning model trees from evolving data streams,” *Data Mining and Knowledge Discovery*, vol. 23, no. 1, pp. 128–168, 2011. <https://doi.org/10.1007/s10618-010-0201-y>.