*Journal of Computer Science*

Research Article

# Adaptive Cross-Validation Under Concept Drift for Time Series Forecasting

**Kunjira Kingphai, Prapakorn Kanjina, Kamol Sanittham and Wacharong Wongsanurak**

*Department of Mathematics and Statistics, Chiang Mai Rajabhat University, Thailand*

*Article history*
Received: 05-11-2025
Revised: 20-05-2026
Accepted: 05-06-2026

**Corresponding Author:**
Kunjira Kingphai
Department of Mathematics and Statistics, Chiang Mai Rajabhat University, Thailand
Email: kunjira@g.cmru.ac.th

**Abstract:** Time-series forecasting often involves non-stationary data, making i.i.d. validation unreliable and fixed-window protocols vulnerable to leakage and biased error estimates. We propose Adaptive Time-Series Cross-Validation (ATSCV), a drift-aware evaluation framework that uses statistical change-point detection to partition each series into contiguous, approximately stationary regimes, followed by forward-chaining folds that respect those boundaries. By aligning train–validation splits with distributional changes (with emphasis on covariate shift), ATSCV yields leakage-controlled, regime-consistent evaluations and more realistic estimates of out-of-sample performance. We evaluate ATSCV on five equity time series (INTC, META, NVDA, ORCL, TSLA) and four model classes (Linear, RNN, LSTM, GRU), using RMSE and MAE. ATSCV reduces RMSE and MAE typically by 30–50% relative to a drift-blind baseline on four of five assets, while revealing one challenging case (TSLA) where frequent regime changes limit cross-regime transfer. Beyond improving accuracy, the protocol stabilizes model rankings and reveals asset-dependent behavior. Overall, the results indicate that drift-aligned evaluation provides more realistic generalization estimates and clarifies when apparent performance is driven by regime dynamics rather than model capability.

**Keywords:** Cross-Validation, Covariate Shift, Time Series, Model Evaluation, Deep Learning

## Introduction

The development of a successful Machine Learning (ML) model is a multi-stage process involving data preprocessing, feature engineering, model training, and evaluation (Ayodele, 2010). Among these stages, training and evaluation are pivotal in ensuring that the model generalizes reliably to unseen data. This is typically assessed via Cross-Validation (CV), which divides the dataset into training and validation subsets. Classical validation techniques such as random train/test (holdout) splits and k-fold cross-validation are based on the assumption that data points are independent and identically distributed (i.i.d.) (Browne, 2000). Formally, i.i.d. implies that each observation $X_1$,..., $X_n$ is independent such that one sample provides no information about another and identically distributed, meaning all samples are drawn from the same underlying distribution (Gnedenko, 2018). However, this assumption is frequently violated in time-series data, owing to temporal dependence and non-stationarity (Hamilton, 2020). Applying shuffle-based cross-validation to time-ordered data can introduce information leakage, wherein training may use future observations while validation relies on past data. This biases performance estimates upward and may result in underperformance during deployment (Bergmeir and Benítez, 2012).

To mitigate these biases, Time-Series Cross-Validation (TSCV) protocols such as expanding and rolling windows preserve temporal order by training on historical segments and validating on subsequent data (Bergmeir et al., 2018). Although these methods are more robust than i.i.d. splits, they remain susceptible to covariate shift a form of non-stationarity in which the feature distribution $P_t(X)$ changes over time, while the conditional distribution $P_t(Y|X)$ remains stable (Kim et al., 2022; Gama et al., 2014). This problem arises across a range of application domains, including finance (Rapach and Zhou, 2020; Taleblou, 2025), energy forecasting (Gasparin et al., 2022), healthcare (Vollmer et al., 2021), and user behavior modeling (Kingphai and Moshfeghi, 2022).

Despite their time-aware structure, standard rolling or expanding windows are often agnostic to the underlying shift structure. Because they rely on fixed window lengths, folds may inadvertently span multiple regimes, masking recent drifts and resulting in unstable performance estimates that vary significantly based on the arbitrary choice of window start-time or length (Cerqueira et al., 2020; Kingphai and Moshfeghi, 2022). These limitations suggest that evaluation frameworks should adapt to detected covariate shifts rather than treating windowing as a static design choice.

In recent years, adapting learning models to nonstationary environments has attracted significant attention. While traditional drift adaptation relied on statistical monitoring or windowing techniques such as adaptive windowing (ADWIN) and Drift Detection Method (DDM), the rise of deep learning has introduced frameworks that address drift at the representation level (Xiang et al., 2023; Yuan et al., 2022). Systematic reviews emphasize the shift toward adaptive neural architectures capable of continual learning and explainable drift detection, particularly in specialized fields such as energy forecasting (Samarajeewa et al., 2024; Abdullahi et al., 2025). Although these studies underscore the trend toward adaptive modeling, they also highlight the lack of a corresponding adaptive evaluation framework a gap that directly motivates the proposed Adaptive Time-Series Cross-Validation (ATSCV) framework. By integrating statistical change-point detection with forward-chaining folds, ATSCV partitions the series into distinct regimes, ensuring that validation aligns with detected shifts. Our main contributions are as follows:

1. We propose a statistically grounded ATSCV protocol that segments the series at detected change points and adapts window sizes and step lengths to the observed non-stationarity
2. We provide a formal analysis of the upward bias introduced by drift-blind validation, along with diagnostic measures for identifying regime-specific performance degradation
3. We demonstrate the use of ATSCV to evaluate model adaptability to covariate shift in a forecasting context. Specifically, we benchmark a linear regression model against a suite of sequence models the standard RNN, the LSTM (Sherstinsky, 2020), and the GRU (Dey and Salem, 2017) on volatile equity data

## *Related Work*

Model evaluation in non-stationary environments is a well-studied yet unresolved challenge. Standard CV techniques, originally designed for i.i.d. data, are known to fail in such settings, prompting the development of specialized time-series validation protocols. However, these protocols often rely on assumptions that are themselves violated by the very data they aim to evaluate.

This section surveys the landscape of evaluation methods, establishing the rationale for a new class of adaptive frameworks. We begin by outlining the standard protocols used in time-series validation and then highlight their critical vulnerability to covariate shift. Subsequently, we examine how most of the existing literature has focused on adapting models rather than improving the evaluation process itself an oversight with significant methodological implications. The section concludes by identifying a key research gap that the proposed work aims to address.

### *Standard Time-Series Validation Protocols*

The standard approach to evaluating models on temporally ordered data is TSCV. Unlike classical CV, which assumes i.i.d. samples and relies on random shuffling (Kohavi, 1995), TSCV preserves temporal integrity by ensuring that validation data always follows training data. This design mimics realistic forecasting scenarios and prevents data leakage from future observations into the training process. Foundational TSCV protocols include the expanding window, rolling window, and blocked cross-validation each designed to respect the data's sequential nature (Bergmeir and Benítez, 2012). Specialized variants, such as Purged and Embargoed CV, have also been developed for financial data to address complex autocorrelation structures (Lainder and Wolfinger, 2022). While these protocols effectively preserve temporal order, their performance is highly sensitive to a critical hyperparameter, namely the training and validation window sizes.

### *Covariate Shift in Time-Series Validation*

The primary reason traditional TSCV protocols struggle is concept drift particularly the frequent and disruptive problem of covariate shift. In real-world data, the statistical properties of the input variables (X) are rarely stationary a phenomenon in which the input distribution changes over time ($P_s(X) \neq P_t(X)$), even if the underlying input–output relationship ($P(Y|X)$) remains stable (Sugiyama et al., 2007).

When a significant covariate shift occurs between the training and validation sets, the training data is no longer representative of the data on which the model is evaluated. A model trained on a pre-shift regime will often perform poorly on a post-shift regime. As a result, any validation score that averages errors across shift boundaries is likely to be misleading, often yielding an overly optimistic estimate of the model's true generalization ability. This problem makes the selection of a fixed window size a critical point of failure: A window that is too long may

average over distinct regimes, while one that is too short may fail to capture stable patterns (Bifet and Gavalda, 2007; Gama et al., 2014).

### *Existing Approaches to Model Adaptation*

In response to the challenges posed by drift, the research community has largely focused on developing adaptive models. These strategies modify the learning algorithm so that it can respond to changes in the data stream. These strategies can be divided into two types. The first is passive adaptation, in which the model is continuously updated over time using mechanisms such as rolling-window regression. The second is active adaptation, in which a drift detector such as ADWIN explicitly identifies a change point, which then triggers a model update or reset (Bifet and Gavalda, 2007). Other common approaches include online learning ensembles and prequential evaluation, where the model is updated and evaluated one data point at a time (Tsymbal, 2004).

### *Evaluation Adaptation in Non-Stationary Settings*

Our review of the literature reveals a persistent disconnection between the methods used for model adaptation and those used for model evaluation. Although numerous studies in high-stakes domains such as finance, clinical prediction, and energy forecasting have documented the performance degradation caused by drift (Li et al., 2022), they typically continue to rely on drift-agnostic rolling or expanding window evaluation schemes.

This highlights a critical gap: While research has extensively addressed model adaptation to handle non-stationarity, the evaluation protocols used to measure success have remained largely static. Traditional rolling-window and prequential methods rely on fixed-step increments that are independent of the data's underlying structure. This lack of alignment can lead to regime-spanning folds, where a single validation set contains data from two distinct distributions. Such misalignment results in an averaged error metric that obscures the model's true failure points a limitation often referred to as evaluation lag. This gap reveals three key research needs:

1. A cross-validation framework that dynamically aligns fold boundaries with statistically detected regime changes, moving beyond fixed or arbitrary window lengths
2. A method to formally characterize and mitigate the optimistic bias introduced by drift-blind evaluation strategies
3. A robust offline validation approach explicitly designed for non-stationary conditions, distinct from online monitoring protocols

ATSCV fundamentally differs from standard approaches by treating the validation structure itself as a dynamic variable. Rather than updating the model to fit the drift, ATSCV adaptively structures the validation folds to align with detected change points. This ensures that each fold represents a statistically coherent regime, allowing researchers to evaluate not only average performance but also the model's specific vulnerabilities to structural breaks.

## Methods

To evaluate model performance under non-stationary conditions, we propose the ATSCV framework. ATSCV integrates the ADWIN algorithm as a drift detector to partition a time series into contiguous, approximately stationary segments. Unlike conventional validation schemes that assume a static data distribution, ATSCV dynamically aligns fold boundaries with statistically detected change points, ensuring that no training or validation fold spans a regime shift. By constructing leakage-controlled, temporally consistent folds, ATSCV yields more reliable out-of-sample error estimates under evolving data distributions. We validate the proposed framework on financial time series and compare its performance against a standard drift-blind baseline.

### *Algorithm Parameters and Initialization*

Prior to execution, the ATSCV framework requires several user-specified parameters and an initialization procedure to ensure consistent and reproducible operation. This subsection outlines the necessary inputs and the initial setup process.

$T$ (time series). The complete, chronologically ordered sequence $T = \{t_1,...,t_n\}$, where $N$ is the number of observations.

$x_i$ (monitored values). A sequence of monitored variables, such as prediction errors or feature residuals. In this study, we monitor the normalized squared error to identify shifts in model performance.

$s_{min}$ (minimum fold size). An integer specifying the minimum allowable segment length. Rather than using a global constant, $s_{min}$ is determined by the architectural requirements of the underlying model. For high-capacity models such as LSTMs and GRUs, $s_{min}$ is set to exceed the model's look-back window and to provide a sufficient number of samples for gradient-based optimization to converge. This prevents the cold-start problem and ensures that each fold constitutes a viable training or validation set.

$\delta$ (drift confidence level). The statistical confidence level for ADWIN. Based on Hoeffding bounds, $\delta$ controls the probability of false alarms. A lower value (e.g., 0.001) makes the detector more conservative, triggering new folds only in the presence of major structural breaks. A

higher value (e.g., 0.1) increases sensitivity to subtle, incremental shifts.

The ADWIN window W is initialized as empty, and a pointer FoldStartIndex is set to the first observation ($i$ = 1). An empty list Folds is also initialized to store the resulting segments.

### *Iterative Segmentation Using ADWIN*

ADWIN is a statistically grounded, adaptive-window algorithm designed to detect changes in data streams. It maintains a variable-length window W of recent observations and continuously tests whether the average values in the older and newer portions of the window differ beyond a statistically defined threshold. When this difference exceeds the ADWIN bound, a drift is detected, and a new segment is initiated.

The segmentation procedure proceeds as follows:

1. For each incoming observation $t_i$ with monitored value $x_i$, append $x_i$ to the ADWIN window W
2. ADWIN continuously monitors the evolving window W by partitioning it into two adjacent sub-windows: $W_0$ (older observations) and $W_1$ (newer observations). It then tests whether a statistically significant difference exists between their respective means:

$$\left|\mu_{W_1} - \mu_{W_0}\right| > \varepsilon_{\text{ADWIN}} \tag{1}$$

The threshold $\varepsilon_{ADWIN}$ is computed using Hoeffding's inequality as:

$$\varepsilon_{\text{ADWIN}} = \sqrt{\frac{1}{2m} 1n\left(\frac{4n}{\delta}\right)} \tag{2}$$

Where $n_0$ and $n_1$ denote the lengths of $W_0$ and $W_1$ Respectively, $n = n\,o + n_1$, and $m = \frac{non_1}{n}$ represents their harmonic mean. If the inequality holds, ADWIN concludes that a distributional change has occurred and contracts the window accordingly.

3. When a drift is detected, ADWIN removes $W_0$ (i.e., $W \leftarrow W_1$) and records a change point at the first index of $W_1$, denoted as $I_{drift}$
4. A candidate fold is then formed, spanning from FoldStartIndex to $I_{drift} - 1$. If the candidate segment length is greater than or equal to $s_{min}$, it is appended to Folds; otherwise, it is discarded and merged into the next regime. The variable FoldStartIndex is updated to $I_{drift}$, marking the start of a new segment

After the final observation $t_N$ is processed, the remaining segment from FoldStartIndex to $t_N$ is appended to Folds, provided it meets the minimum size requirement $s_{min}$. This process yields a sequence of contiguous folds that adaptively align with the change points detected by ADWIN.

### *ADWIN Foundation and Statistical Guarantee*

ADWIN provides theoretical guarantees for controlling false positive detections when monitoring bounded variables. For binary or bounded values $x_i \in [0,1]$, Hoeffding's inequality ensures that the probability of falsely declaring a drift (Type I error) is bounded by δ. As a result, ATSCV inherits ADWIN's probabilistic reliability while transforming detected change points into meaningful cross-validation folds that preserve both temporal and distributional integrity.

Figure 1 illustrates the two-stage ATSCV framework. In the upper panel, the time series is continuously monitored by the ADWIN drift detector, which maintains a variable-length sliding window and identifies statistically significant changes in the underlying data distribution. When ADWIN detects a drift defined as a mean difference between adjacent sub-windows that exceeds the Hoeffding bound $\varepsilon_{ADWIN}$ a change point is inserted, and all preceding observations are consolidated into a stable segment (fold). Consecutive observations without detected drift remain within the same window, producing adaptively sized folds that approximate local stationarity within each regime.

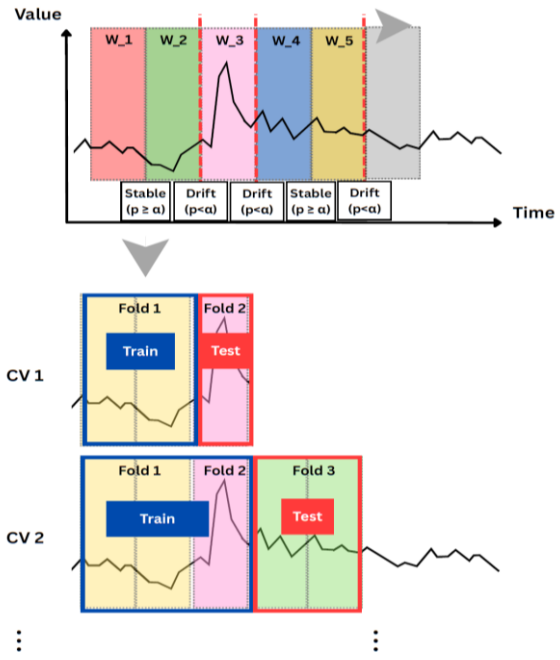


**Fig. 1:** Adaptive Time-Series Cross-Validation (ATSCV)

In the lower panel, these folds are used to construct leakage-controlled, temporally ordered cross-validation splits. For example, Split 1 trains on Fold 1 and validates on Fold 2; Split 2 trains cumulatively on Folds 1–2 and validates on Fold 3, and so on. No split crosses a detected change point, ensuring that model evaluation respects both temporal order and regime boundaries.

Although ATSCV is modular and supports other detectors such as Pruned Exact Linear Time (PELT) (Killick et al., 2012) or the cumulative sum (CUSUM) algorithm (Page, 1954) ADWIN was selected for its online adaptability and lack of fixed look-back requirements. An empirical comparison of ADWIN, PELT, and CUSUM within the ATSCV framework is provided in Section Detector Comparison, where we discuss the performance–operability trade-off between detectors in both offline and streaming settings.

Together, the schematic in Figure 1 and the discussion of detector modularity provide a conceptual overview of how ATSCV partitions non-stationary time series into statistically stable folds. The next subsection formalizes this framework as a step-by-step algorithm that integrates ADWIN-based drift detection with adaptive fold construction for TSCV.

### *Algorithm Description*

The overall ATSCV procedure integrates ADWIN-based drift detection with adaptive fold segmentation for time-series cross-validation. The algorithm incrementally monitors a stream of feature or error values, applies ADWIN's statistical test for distributional change, and partitions the time series into non-overlapping folds corresponding to locally stationary regimes. Each detected drift boundary marks the end of one fold and the beginning of the next, ensuring that no training or validation set spans a detected shift.

Note: The routine ADWIN_Contract(W, δ) tests all possible splits of W using $\varepsilon_{ADWIN}$ (Equation 2). When a drift is detected, it discards the oldest sub-window $W_0$ from W, retaining only $W_1$, and returns the discarded segment, $W_{old}$. If no drift is found, it returns an empty set.

**Algorithm 1** ADWIN-Based Segmentation for Adaptive Time-Series Cross-Validation (ATSCV)

**Require:** Monitored stream $X = [x_1,...,x_n]$ (e.g., error indicators); ADWIN confidence δ; minimum fold size $s_{min}$
**Ensure:** Set of contiguous folds $F = \{Fold_1, Fold_2, ...\}$
Initialize ADWIN window $W \leftarrow \emptyset$
Initialize $F \leftarrow \emptyset$; $FoldStart \leftarrow 1$
**for** i = 1 to n **do**
  Append $x_i$ to W
  $W_{old} \leftarrow$ ADWIN_Contract(W, δ) {Returns discarded portionW0; empty if no drift}
  **If** $W_{orld}$ is NOT empty **then**
  $DriftIndex \leftarrow i - |W| + 1$ {First index of the new regimeW1 in the original stream}
    $FoldEnd \leftarrow DriftIndex - 1$
    **if** $FoldEnd - FoldStart + 1 \geq s_{min}$ **then**
      $Fold_{new} \leftarrow X[FoldStart : FoldEnd]$
      Add $Fold_{new}$ to F
    **end if**
    $FoldStart \leftarrow DriftIndex$ {Advance start regardless—discard short fold or begin new regime}
  **end if**
**end for**
**if** $n - FoldStart + 1 \geq s_{min}$ **then**
  $Fold_{final} \leftarrow X[FoldStart : n]$
  Add $Fold_{final}$ to *F*
**end if**
**return** *F*

### *Data and Feature Engineering*

This study uses daily stock price data for five large-cap, highly liquid U.S. equities: Intel Corporation (INTC), Meta Platforms, Inc. (META), NVIDIA Corporation (NVDA), Oracle Corporation (ORCL), and Tesla, Inc. (TSLA). These assets were selected to capture a range of volatility dynamics and trading behaviors across different sectors. The dataset was obtained from Yahoo Finance via the yfinance library and spans a ten-year period from January 2, 2015, to December 31, 2024. For each stock, the variables Close, High, Low, and Volume were collected. After validating data completeness and resolving missing values, a set of causal and interaction-based predictors was developed.

The simple daily return $r_t$ and the seven-day rolling volatility $\sigma_t^{(7)}$ are defined as:

$$r_t = \frac{C_t - C_{t-1}}{C_{t-1}} \tag{3}$$

$$\sigma_t^{(7)} = sd(r_{t-6},...,r_t) \tag{4}$$

$\sigma_t^{(7)} = sd(r_{t-6},...,r_t)$ recent seven observations. To account for trading intensity and nonlinear effects between price movements and volume, two additional variables are derived: the logarithm of trading volume $\ell_t = log\ V_t$ and an interaction term between return and log-volume, $r_t \times \ell_t$. The resulting feature vector at each time step t is expressed as:

$$x = [r,\ \sigma^{(7)},\ \ell,\ r \times \ell]^7 \tag{5}$$

The prediction target corresponds to the contemporaneous closing price:

$$y_t = C_t \tag{6}$$

While financial forecasting typically focuses on

forward horizons (e.g., $C_{t+1}$), we intentionally model the contemporaneous price to isolate the relationship between observed market dynamics and the contemporaneous price level. This choice is specifically designed to facilitate the evaluation of the ATSCV framework; by focusing on the current price, we minimize the predictive noise and the lead-lag variance inherent in future-dated returns. This allows for a more precise identification of when the statistical relationship between features and targets breaks down, providing a rigorous environment in which to test the framework's ability to detect regime shifts and align validation folds accordingly.

### *Forecasting Models*

We evaluate a baseline linear model and three recurrent neural network (RNN) architectures that are commonly used for sequence modeling. A standard linear regression model (Ordinary Least Squares) is trained on standardized features, with z-score normalization fitted on the training data within each fold. We implement vanilla RNN, LSTM, and GRU architectures using a common design to ensure a fair comparison across recurrent cells. All neural models are trained using the Adam optimizer (learning rate = $10^{-3}$) to minimize mean squared error (MSE). Early stopping, with a patience of 10 epochs, monitors the validation loss within each fold. The shared architecture is summarized in Table 1.

### *Evaluation Protocol*

Two CV strategies are employed to evaluate the forecasting models: A conventional, drift-unaware baseline and the proposed, drift-aware ATSCV method.

Baseline: Fixed Rolling Window Method. The Baseline applies a fixed rolling window CV, a standard timeseries procedure that maintains temporal order but ignores potential regime shifts. Each training fold comprises a fixed-length window immediately preceding the validation fold. Two configurations are considered:

(1) A fixed fivefold setup ($K_5$)
(2) A variable-fold setup ($K_x$), in which the number of folds equals the number of change points detected by the adaptive ATSCV procedure for each asset

This design enables a direct fold-by-fold comparison of the conventional and adaptive approaches under equivalent data partitions.

**Table 1:** Neural network architecture shared by RNN, LSTM, and GRU models

| Layer | Type / Activation | Units / Rate |
|---|---|---|
| 1 | Recurrent (RNN / LSTM / GRU) | 50 units |
| 2 | Dropout | 0.3 |
| 3 | Recurrent (RNN / LSTM / GRU) | 25 units |
| 4 | Dense (ReLU) | 25 units |
| 5 | Output (Linear) | 1 unit |

Proposed Method: ATSCV. The ATSCV procedure, described in Section of *Algorithm Description* 3, integrates the ADWIN drift detector to segment the series into statistically homogeneous regimes and applies forward-chaining validation across these detected segments. Unlike the baseline, fold boundaries in ATSCV are determined dynamically by statistically significant distributional changes rather than by fixed intervals. This alignment ensures that each train–validation split respects the underlying data-generating structure, thereby minimizing leakage across regime shifts and yielding a more realistic estimate of generalization performance.

The two evaluation protocols described above were applied consistently across all assets and model architectures. This design enables a controlled comparison of conventional drift-blind validation and the proposed drift-aware ATSCV framework. By evaluating models under both protocols, the analysis isolates the effect of adaptive, drift-aligned segmentation on forecasting accuracy and stability. The following section presents the empirical results, highlighting how drift-aware evaluation influences model robustness and error behavior under non-stationary financial time series.

### *Performance Metrics*

Model performance is assessed using two standard regression metrics, computed for each CV fold and averaged across all folds.

Root Mean Squared Error (RMSE). For a given fold $f$ containing $n_f$ observations and corresponding predictions $\hat{y}_t$, the RMSE is defined as:

$$\mathrm{RMSE}_f = \sqrt{\frac{1}{n_f}\sum_{t=1}^{n_f}(y_t \hat{} y)^2} \tag{7}$$

Mean Absolute Error (MAE). The MAE for the same fold is calculated as:

$$MAE_f = \frac{1}{n_f}\sum_{t=1}^{nf}\left|y_{i-} y_t\right| \tag{8}$$

The fold-level metrics are then aggregated across all $F$ folds using simple arithmetic means:

$$\overline{\mathrm{RMSE}} = \frac{1}{F}\sum_{f=1}^{F}\mathrm{RMSE}_f, \overline{MAE} = \frac{1}{F}\sum_{f-1}^{F}MAE_f \tag{9}$$

RMSE is sensitive to large deviations due to its quadratic penalty on errors, making it suitable for evaluating models where high-magnitude prediction failures are costly. MAE, in contrast, provides a more interpretable measure of typical forecast error by penalizing deviations linearly. The joint use of both

metrics enables a balanced assessment of predictive accuracy and error stability across models and validation protocols.

Given this study's focus on evaluating model robustness under concept drift, the primary comparison emphasizes results obtained through the proposed ATSCV protocol. The source code implementing the ATSCV procedure is available upon request.

## Results and Discussion

This section presents the empirical evaluation of the ATSCV framework in three stages: Detector selection, parameter optimisation, and main experimental results. We begin by comparing three drift detectors to motivate the choice of ADWIN, then determine its optimal configuration through sensitivity analysis, before reporting the main findings across all assets, models, and domains.

### *Detector Comparison*

To validate the modularity of ATSCV and motivate the selection of ADWIN as the primary detector, we compared three detector configurations within the same pipeline: ADWIN ($\delta = 0.05$), PELT (penalty = 50), and CUSUM ($k = 0.5$, $h = 3.0$). All detectors monitored the same normalised residual stream, used identical fold construction and model training procedures, and were parameterised to produce a comparable number of folds (~5–8) per asset. The $K_5$ fixed rolling window serves as the baseline.

All three detectors produce valid drift-aligned fold structures, confirming the framework's modularity.

MAE results follow the same pattern and are reported in Table 2 (Fig. 2).

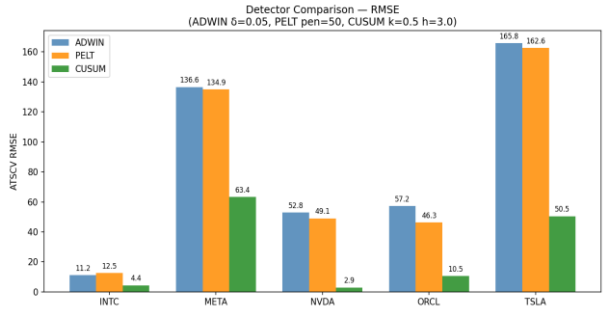


**Fig. 2:** Mean RMSE per asset for ADWIN, PELT, and CUSUM within the ATSCV pipeline

**Table 2:** Detector comparison averaged across all five assets. Imp. = % improvement over K5 baseline. RT = segmentation runtime (ms)

| Detector | RMSE | MAE | Imp. | RT (ms) |
|---|---|---|---|---|
| ADWIN ($\delta = 0.05$) | 84.73 | 81.60 | −112.1 | 713.2 |
| PELT (pen = 50) | 81.07 | 78.53 | −97.9 | 3175.9 |
| CUSUM ($k = 0.5$, $h = 3.0$) | 26.34 | 22.47 | +49.6 | 1.2 |

CUSUM achieves the lowest errors (RMSE 26.3, MAE 22.5) and the highest improvement over baseline (+49.6%), while ADWIN and PELT produce comparable results. However, CUSUM relies on fixed global statistics precomputed from the full series, making it a batch-oriented detector unsuitable for online deployment, as evidenced by its near-zero runtime. ADWIN was therefore selected as the primary detector for its online adaptability, absence of fixed look-back requirements, and formal probabilistic guarantees on false alarm rates via Hoeffding bounds properties essential in streaming settings. Practitioners in offline batch settings may substitute CUSUM or PELT to leverage their performance advantage within the modular ATSCV pipeline. Having selected ADWIN as the primary detector, we next determine its optimal parameter configuration through a systematic sensitivity analysis.

### *Sensitivity Analysis*

To evaluate the robustness of the ATSCV framework, we conducted a systematic sensitivity analysis over a grid of 30 parameter combinations, varying the ADWIN confidence level $\delta \in \{0.0001, 0.001, 0.002, 0.01, 0.05, 0.1\}$ and the minimum fold size $s_{min} \in \{30, 60, 90, 120, 150\}$. For each configuration, ATSCV was applied to all five assets and four model architectures, and the resulting RMSE and MAE were averaged to obtain aggregate estimates. Tables 3 and 4 report the full results, and Figure 3 shows the corresponding mean fold counts.

The results show a clear monotonic improvement as $\delta$ increases from 0.0001 to 0.1: Mean RMSE falls from approximately 143 to 116, and mean MAE from 140 to 113.

The effect of $s_{min}$ is most pronounced at higher $\delta$ values, where larger minimum fold sizes filter spurious detections and further reduce error.

**Table 3:** Mean RMSE across all assets and models for each ($\delta$, $s_{min}$) combination. Best result in bold

| $\delta$ | $s_{min}$ | | | | |
|---|---|---|---|---|---|
| | 30 | 60 | 90 | 120 | 150 |
| 0.0001 | 143.30 | 143.30 | 139.23 | 139.23 | 139.23 |
| 0.001 | 141.19 | 141.19 | 141.20 | 141.20 | 148.05 |
| 0.002 | 141.26 | 141.26 | 141.25 | 141.45 | 141.45 |
| 0.01 | 121.85 | 121.85 | 126.01 | 126.19 | 124.69 |
| 0.05 | 122.17 | 122.17 | 122.55 | 122.74 | 122.74 |
| 0.1 | 119.32 | 119.32 | 119.64 | 119.83 | 116.07 |

**Table 4:** Mean MAE across all assets and models for each ($\delta$, $s_{min}$) combination. Best result in bold

| $\delta$ | $s_{min}$ | | | | |
|---|---|---|---|---|---|
| | 30 | 60 | 90 | 120 | 150 |
| 0.0001 | 139.85 | 139.85 | 135.70 | 135.70 | 135.70 |
| 0.001 | 138.08 | 138.08 | 137.99 | 137.99 | 144.20 |
| 0.002 | 138.18 | 138.18 | 138.05 | 138.25 | 138.25 |
| 0.01 | 117.24 | 117.24 | 121.39 | 121.58 | 119.32 |
| 0.05 | 117.74 | 117.74 | 118.11 | 118.30 | 118.30 |
| 0.1 | 115.70 | 115.70 | 116.02 | 116.20 | 112.53 |

**DOI:** 10.3844/jcssp.2026.2104.2117

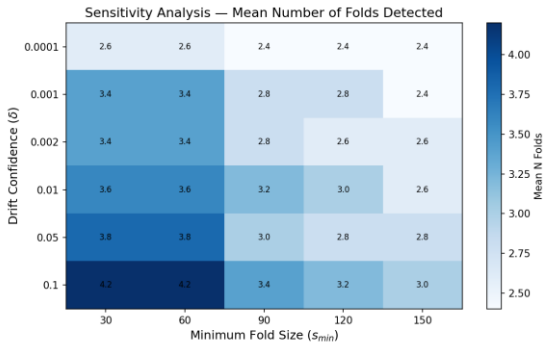


**Fig. 3:** Mean number of folds detected across the sensitivity grid. Higher δ and smaller $s_{min}$ produce more folds

The combination δ = 0.1, $s_{min}$ = 150 achieves the lowest RMSE (116.07) and MAE (112.53) and was therefore selected as the default parameterization for all main experiments reported hereafter.

Using ADWIN with the optimal configuration (δ =0.1, $s_{min}$ = 150), we now present the main experimental results. The following analysis employs the Kx baseline, in which the number of folds matches those detected by ATSCV for each asset, enabling a direct fold-for-fold comparison.

### *Main Experimental Results*

We compare Baseline vs. ATCV per model on each stock. Percent improvement is computed as:

$$\text{Improvement}(\%) = 100 \times \frac{\text{Baseline} - \text{Proposed}}{\text{Baseline}}$$

As shown in Table 5, ATSCV reduces both RMSE and MAE relative to baseline cross-validation for four of the five equities across all model classes.

The largest improvements occur for the linear model, while recurrent architectures also benefit, albeit to a lesser extent. An exception is TSLA, for which all models exhibit higher errors under ATSCV-likely due to frequent or severe regime changes that shorten effective training histories and limit transferability across regimes. Taken together, these findings suggest that drift-aligned evaluation offers a more conservative and informative estimate of out-of-sample performance—strengthening conclusions for series with moderate drift while highlighting assets whose dynamics are dominated by rapid structural change.

Figure 4 quantifies the percentage change in RMSE and MAE relative to baseline CV and supports the tabular findings.

**Table 5:** Baseline vs. ATSCV (Adaptive Time-Series Cross-Validation)

| Stock | Model | RMSE | | MAE | |
|---|---|---|---|---|---|
| | | Baseline | ATSCV | Baseline | ATSCV |
| INTC | GRU | 14.318 | 11.522 | 13.007 | 10.724 |
| | LSTM | 14.993 | 12.638 | 13.697 | 11.816 |
| | Linear | 19.743 | 11.548 | 18.109 | 10.574 |
| | RNN | 17.235 | 12.445 | 15.710 | 11.655 |
| META | GRU | 38.480 | 23.385 | 34.068 | 21.074 |
| | LSTM | 43.985 | 25.031 | 38.080 | 22.384 |
| | Linear | 62.270 | 30.125 | 57.953 | 27.830 |
| | RNN | 42.719 | 26.081 | 38.349 | 23.642 |
| NVDA | GRU | 9.333 | 5.594 | 8.814 | 5.231 |
| | LSTM | 10.196 | 6.732 | 9.732 | 6.271 |
| | Linear | 10.776 | 6.058 | 9.468 | 5.524 |
| | RNN | 11.272 | 6.367 | 10.753 | 6.136 |
| ORCL | GRU | 12.826 | 7.177 | 11.077 | 6.641 |
| | LSTM | 13.173 | 8.184 | 11.371 | 7.719 |
| | Linear | 18.339 | 9.033 | 17.136 | 8.633 |
| | RNN | 15.651 | 7.969 | 14.279 | 7.437 |
| TSLA | GRU | 13.887 | 24.189 | 10.986 | 22.397 |
| | LSTM | 16.721 | 30.294 | 13.453 | 28.281 |
| | Linear | 19.893 | 29.727 | 16.732 | 25.862 |
| | RNN | 19.268 | 28.491 | 15.392 | 25.872 |

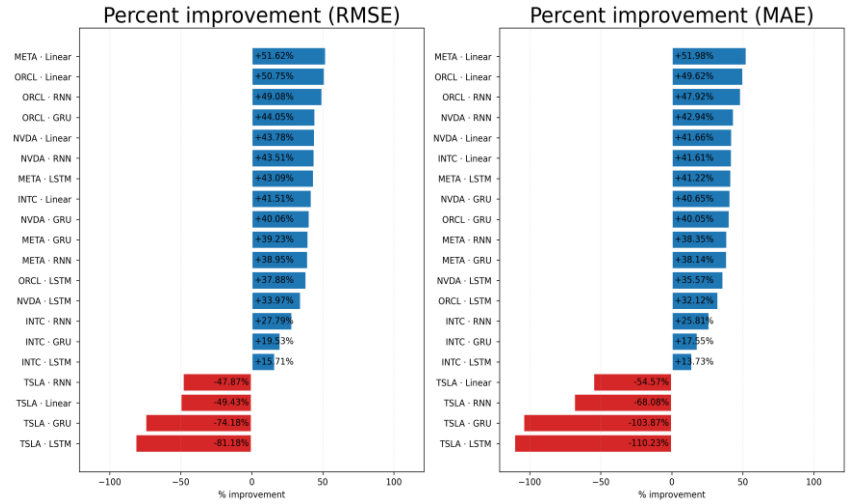


**Fig. 4:** Percentage improvement in RMSE and MAE for the ATSCV method relative to the Baseline. Positive values (blue) indicate improved performance, while negative values (red) indicate degraded performance

The pattern is clearly asset-dependent: For INTC, META, NVDA, and ORCL, all models improve with typical gains around 30–50%, and several combinations exceed 50% (e.g., META–Linear and ORCL–Linear). In contrast, TSLA deteriorates across models, with some increases exceeding 100%. Overall, the figure shows that ATSCV reduces error for four of the five assets while its effectiveness remains conditional on regime structure, underscoring a pronounced method–by–asset interaction. To assess the impact of the proposed validation protocol across model architectures, we analyzed the average prediction error across all assets for each of the four models: GRU, LSTM, linear regression, and a standard RNN.

Figures 5 and 6 illustrate the average RMSE and MAE for models evaluated using both the baseline CV and the proposed ATSCV. The results reveal two consistent and important findings.

First, the ATSCV method produced a substantial reduction in average error across all tested models. As shown in both charts, the average RMSE and MAE for models evaluated with ATSCV (blue bars) are consistently lower than those obtained with the baseline CV (orange bars), indicating that the drift-aware protocol provides a more effective training and validation framework regardless of model architecture.

Second, the linear model exhibited the highest initial error under the baseline CV and subsequently achieved the largest absolute and relative reductions in both RMSE and MAE when evaluated with ATSCV.

This finding suggests that simpler models, which are less capable of internally capturing non-stationarities, benefit most from a validation strategy that explicitly segments data into stable regimes.

In summary, these results demonstrate that the advantages of the ATSCV framework are consistent across multiple common time-series models and are particularly beneficial for simpler, linear architectures. To further assess the performance of the ATSCV protocol across different financial instruments, we also analyzed the average prediction error for each stock, aggregated across all models.

Figures 7 and 8 present the average RMSE and MAE for each stock under both the baseline and ATSCV evaluation methods. The results reveal two distinct and critical patterns.

For the majority of stocks tested including INTC, META, NVDA, and ORCL the ATSCV framework yielded substantial reductions in prediction error. The consistently lower RMSE and MAE values for these assets demonstrate the framework's effectiveness in contexts where drift-aware segmentation enhances model stability and generalization.

In contrast, the ATSCV method resulted in a marked performance decline for TSLA, with both RMSE and MAE notably higher than those under the baseline CV.

These divergent outcomes suggest that the effectiveness of the ATSCV protocol depends on the inherent characteristics of each time series. While the method provides a clear advantage for most assets, the TSLA case suggests that for highly volatile or uniquely structured data, a drift-aligned validation scheme may be suboptimal. This insight underscores the absence of a universal evaluation strategy for financial time-series forecasting.

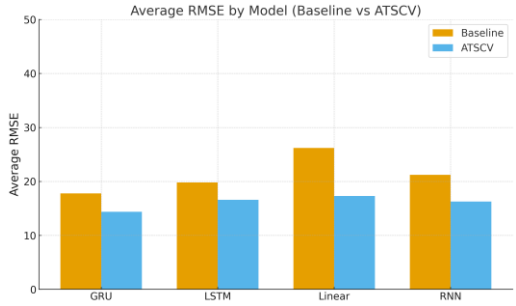


**Fig. 5:** Average RMSE by model under Baseline and ATSCV evaluation methods

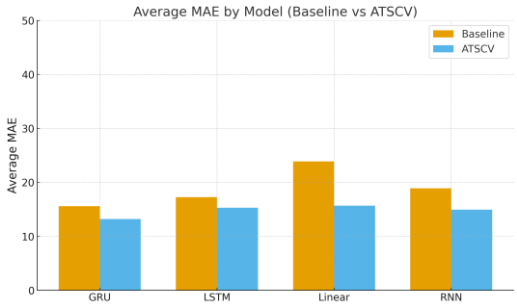


**Fig. 6:** Average MAE by model under Baseline and ATSCV evaluation methods

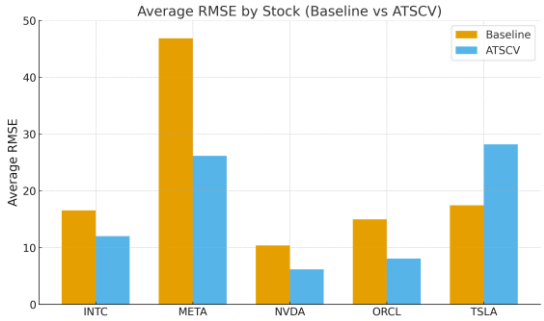


**Fig. 7:** Average RMSE by stock under Baseline and ATSCV evaluation

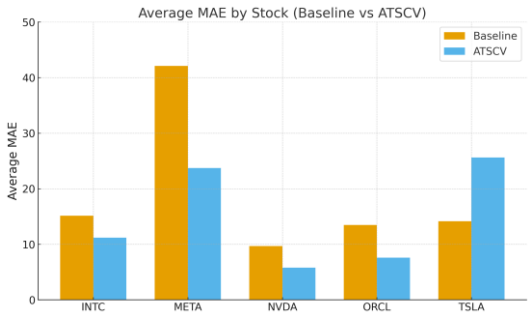


**Fig. 8:** Average MAE by stock under Baseline and ATSCV evaluation

### *Variance Fold Analysis*

While the previous analysis established that ATSCV effectively reduces average error, Figures 9–13 offer a more detailed perspective on model robustness by visualizing the full distributions of prediction errors across all cross-validation folds. These violin plots enable assessment not only of expected accuracy but also of each model's performance stability and risk profile under varying market conditions.

For INTC (Figure 9), the ATSCV protocol yields a visibly narrower and more compact error distribution compared to the baseline, indicating reductions in both median error and overall variance. A similar pattern is observed for META (Figure 10), where the ATSCV violin is shorter with thinner tails, indicating improved model consistency and reduced susceptibility to extreme forecast deviations. The NVDA results (Figure 11) reinforce this trend, exhibiting a significant contraction in the upper tail, implying reduced exposure to high-magnitude errors. For ORCL (Figure 12), the difference between evaluation methods is less pronounced but still favors ATSCV, which yields a smaller spread and a slightly lower median, consistent with improved stability.

In contrast, the TSLA violin (Figure 13) displays the opposite pattern. The ATSCV distribution is wider and more skewed, with a noticeably elongated upper tail, reflecting increased variability and a higher likelihood of large forecast errors relative to the baseline. This divergence suggests that for assets characterized by high volatility and frequent structural shifts, such as TSLA, adaptive segmentation may amplify sensitivity to unstable patterns rather than mitigate them.

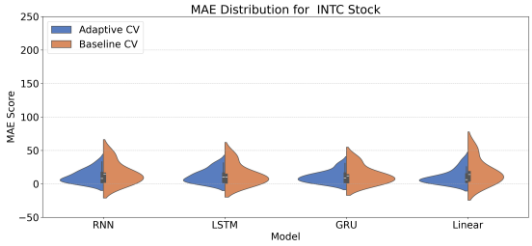


**Fig. 9:** Violin plot of error distributions under Baseline vs. ATSCV for INTC

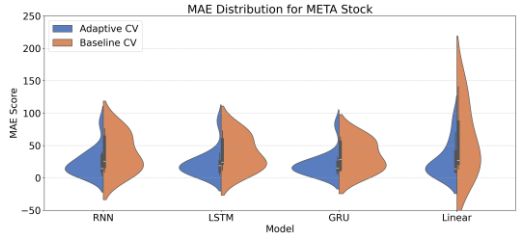


**Fig. 10:** Violin plot of error distributions under Baseline vs. ATSCV for META

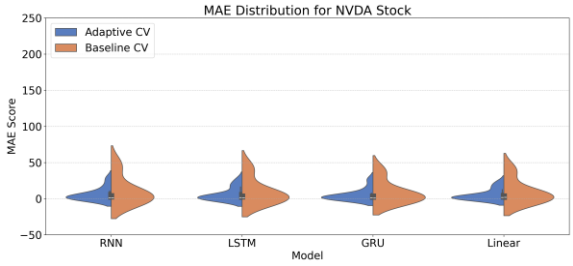


**Fig. 11:** Violin plot of error distributions under Baseline vs. ATSCV for NVDA

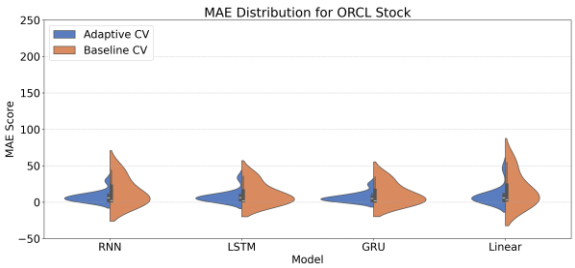


**Fig. 12:** Violin plot of error distributions under Baseline vs. ATSCV for ORCL

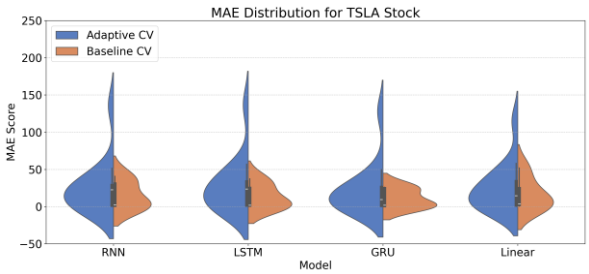


**Fig. 13:** Violin plot of error distributions under Baseline vs. ATSCV for TSLA

Overall, these visual comparisons demonstrate that ATSCV not only improves average performance for most assets but also reshapes the underlying error structure, resulting in models that are both more accurate and more robust to distributional noise. The exception of TSLA underscores that the effectiveness of drift-aware validation is inherently asset-dependent, reflecting the distinct temporal dynamics and regime behaviors of individual time series.

### *Statistical Hypothesis Testing*

To formally assess the observed effects, a three-way analysis of variance (ANOVA) was conducted. The analysis evaluated the impact of the validation method (CV Method), the forecasting model (Model), and the specific stock (Stock) on prediction error, as measured by RMSE. The results, shown in Table 6, provide statistical support for our main findings.

**Table 6: Three-way ANOVA on RMSE comparing CV Method, Model, and Stock**

| Effect | df | F | p | Sig. (p<0.05) |
|---|---|---|---|---|
| CV Method | 1 | 3.94 | 0.048* | Yes |
| Model | 3 | 0.61 | 0.606 | No |
| Stock | 4 | 16.11 | <0.001*** | Yes |
| CV Method × Model | 3 | 0.26 | 0.854 | No |
| CV Method × Stock | 4 | 3.67 | 0.006** | Yes |
| Model × Stock | 12 | 0.16 | 1.000 | No |
| CV Method × Model × Stock | 12 | 0.08 | 1.000 | No |

Notes: Stars denote significance levels: ∗ p<0.05, ∗∗ p<0.01,∗∗∗ p<0.001. Residual df = 324 (not shown to simplify layout)

The analysis revealed a statistically significant main effect of CV Method ($F(1,324) = 3.94$, $p = 0.048$), confirming that our proposed ATSCV protocol led to a significant reduction in prediction error compared to the baseline. A significant main effect was also observed for Stock ($F(4,324) = 16.11$, $p < 0.001$), verifying that prediction difficulty varies significantly across financial assets.

Critically, the ANOVA identified a significant CV Method × Stock interaction ($F(4,324) = 3.67$, $p = 0.006$). This interaction provides statistical evidence for a key finding: The performance benefit of the ATSCV method is not uniform, but contingent upon the specific characteristics of each stock.

The main effect of Model was not significant ($F(3,324) = 0.61$, $p = 0.606$), and no other interaction terms reached significance ($p > 0.05$). This reinforces that the critical factor influencing performance gains is not the choice of model, but the interaction between validation method and the unique statistical properties of each asset.

### *Generalisation to Non-Financial Domains*

To assess the applicability of ATSCV beyond the financial domain, we evaluated the framework on two publicly available benchmark datasets: ETTh1 (Zhou et al., 2021), an hourly electricity transformer temperature series (2,475 observations, target: Oil temperature OT) from the energy domain; and the UCI Air Quality dataset (De Vito et al., 2008), an hourly carbon monoxide concentration series (1,999 observations, target: CO(GT)) from the environmental domain. Both datasets exhibit non-stationarity driven by distinct mechanisms smooth seasonal cycles in ETTh1 and episodic pollution events in AirQuality. The $K_5$ baseline and all model architectures are identical to those used in the detector comparison, and metrics are averaged across all four models.

As shown in Table 7, ATSCV outperforms the baseline on AirQuality (+5.4%), where ADWIN detected five drift points aligned with episodic pollution events. For ETTh1, performance is marginally lower (−3.2%), as the smooth seasonal cycle produced only two drift points, limiting the adaptive benefit of drift-aligned segmentation. These results confirm that ATSCV transfers to non-financial domains without modification, and that its effectiveness is conditioned on the nature of the underlying drift.

**Table 7:** Generalisation results. BL = $K_5$ baseline, AT = ATSCV. Imp. = average % improvement over baseline

| Dataset | BL RMSE | AT RMSE | Imp. (%) |
|---|---|---|---|
| ETTh1 | 1.735 | 1.795 | −3.2 |
| AirQuality | 2.456 | 2.319 | +5.4 |

### *Discussion*

The empirical results provide comprehensive validation for our primary contribution: A novel, drift-aligned cross-validation protocol. The consistent and significant reduction in both RMSE and MAE across a majority of assets (Table 6), coupled with the statistical significance confirmed by the ANOVA, directly supports our claim that aligning CV folds with detected change points improves predictive accuracy. Furthermore, the analysis of error distributions shown in Figures 9 to 13 reveals that our method also enhances model stability by mitigating high-magnitude errors. Taken together, this evidence demonstrates that the proposed procedure is not merely a theoretical construct but a practical tool that yields more robust and reliable model assessments under real-world conditions of concept drift.

Our results also validate our second contribution. First, the violin plots serve as the proposed diagnostic tools, enabling clear analysis of segment-specific model degradation and stability. Second, the substantial performance gap between our method and the baseline highlights the optimistic bias inherent in drift-agnostic CV, which fails to capture the true cost of prediction error following a regime change.

Regarding computational overhead, ADWIN operates at O(logn) amortised complexity per observation, yielding O(N logN) total segmentation cost compared to O(N) for a fixed rolling window. In practice, ADWIN completed segmentation in 713ms on average ($N \approx 2,500$) versus 3,176ms for PELT, both of which are negligible relative to neural network training. This O(logn) per-observation complexity and compressed bucket memory also make ATSCV directly applicable to streaming settings, with $s_{min}$ providing a natural scalability control by trading detection resolution for computational efficiency. All data used in this study are publicly available stock prices via yfinance and non-financial benchmarks at the URLs cited in Section Generalisation to Non-Financial Domains and the complete source code is available upon request, with a public repository to be released upon acceptance.

## Conclusion

This paper introduced and validated an ATSCV protocol designed to support more reliable model evaluation in non-stationary environments. Our empirical results demonstrated that this drift-aligned procedure leads to statistically significant improvements in predictive accuracy and model stability. Furthermore,

using the diagnostic tools developed in this work, we identified a crucial, asset-dependent pattern: The protocol's benefits are substantial but contingent on the underlying dynamics of the data.

This approach marks a significant departure from previous research. The majority of existing work focuses on model adaptation designing algorithms like ADWIN (Bifet and Gavaldá, 2007) or advanced deep learning models that can learn from changing data streams (Lu et al., 2018). While valuable, these methods are often assessed with traditional fixed-window validation protocols (Cerqueira et al., 2020) that are blind to the very drifts the models aim to handle. Our work addresses this fundamental gap by focusing on evaluation adaptation. By making the cross-validation protocol itself sensitive to drift, our ATSCV framework provides a more robust and realistic estimate of a model's true generalisation performance, overcoming the optimistic bias inherent in drift-agnostic validation methods.

For practitioners implementing machine learning in volatile domains, we offer three primary recommendations based on our findings:

1. Align Folds to Regimes: In environments prone to structural breaks, practitioners should move away from fixed-step rolling windows toward drift-aligned boundaries to prevent regime-spanning folds from skewing performance metrics
2. Calibrate Detection Sensitivity: The drift confidence parameter (δ) should be tuned according to the model's objective; use conservative values (e.g., 0.001) for long-term strategic models and higher values (e.g., 0.1) for tactical models requiring rapid adaptation, consistent with the sensitivity analysis reported in Section Sensitivity Analysis
3. Use Dynamic Retraining Schedules: ATSCV should be employed as a diagnostic tool to determine optimal retraining frequency, rather than relying on arbitrary time-based updates that may occur too late or too frequently

While validated on financial data and confirmed transferable to energy and environmental domains, ATSCV is domain-agnostic and modular, allowing for the substitution of ADWIN with detectors like PELT or CUSUM to balance precision and speed. Future research should extend evaluation to additional domains such as industrial IoT and clinical monitoring, and explore high-frequency time series to identify the data characteristics that best inform evaluation strategies.

In summary, this study contributes a novel, adaptive cross-validation protocol and a formal analysis of optimistic bias in traditional CV. By prioritising drift-aware evaluation, this work establishes a foundation for more robust performance estimation in non-stationary environments.

## Acknowledgment

The authors would like to express their sincere appreciation to the Department of Mathematics and Statistics, Chiang Mai Rajabhat University, for providing academic support and research facilities that made this work possible. The authors also thank colleagues for their valuable comments during manuscript preparation.

## Funding Information

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

## Author's Contributions

**Kunjira Kingphai:** Conceptualized the research, designed the study, developed the ATSCV framework, provided the initial codebase, conducted the analysis, interpreted the results, wrote the manuscript, and supervised the overall research process.

**Prapakorn Kanjina:** Implemented the experiments and contributed to code development and model evaluation.

**Kamol Sanittham and Wacharong Wongsanurak:** Provided revisions and feedback on the final manuscript. All authors reviewed and approved the final version of the manuscript.

## Ethics

This study did not involve human participants, animal experiments, or sensitive personal data. All data used were publicly available financial time-series records obtained from Yahoo Finance. No ethical concerns are anticipated following the publication of this manuscript.

## References

Abdullahi, M., Alhussian, H., Aziz, N., Jadid Abdulkadir, S., Baashar, Y., Ahmed Alashhab, A., & Afrin, A. (2025). A Systematic Literature Review of Concept Drift Mitigation in Time-Series Applications. *IEEE Access*, *13*, 119380–119410. https://doi.org/10.1109/access.2025.3587231

Ayodele, T. O. (2010). Types of machine learning algorithms *New Advances in Machine Learning*, *3*, 19–48. https://doi.org/10.5772/9385

Bergmeir, C., & Benítez, J. M. (2012). *On the use of cross-validation for time series predictor evaluation*. Information Sciences. https://doi.org/10.1016/j.ins.2011.12.028

Bergmeir, C., Hyndman, R. J., & Koo, B. (2018). A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics & Data Analysis*, *120*, 70–83. https://doi.org/10.1016/j.csda.2017.11.003

Bifet, A., & Gavaldà, R. (2007). Learning from Time-Changing Data with Adaptive Windowing. *Proceedings of the 2007 SIAM International Conference on Data Mining*, 443–448. https://doi.org/10.1137/1.9781611972771.42

Browne, M. W. (2000). Cross-validation methods. *Journal of Mathematical Psychology*, *44*(1), 108–132. https://doi.org/10.1006/jmps.1999.1279

Cerqueira, V., Torgo, L., & Mozetič, I. (2020). Evaluating time series forecasting models: an empirical study on performance estimation methods. *Machine Learning*, *109*(11), 1997–2028. https://doi.org/10.1007/s10994-020-05910-7

De Vito, S., Massera, E., Piga, M., Martinotto, L., & Di Francia, G. (2008). On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario. *Sensors and Actuators B: Chemical*, *129*(2), 750–757. https://doi.org/10.1016/j.snb.2007.09.060

Dey, R., & Salem, F. M. (2017). Gate-variants of Gated Recurrent Unit (GRU) neural networks. *Proceedings of the 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, 1597–1600. https://doi.org/10.1109/mwscas.2017.8053243

Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM Computing Surveys*, *46*(4), 1–37. https://doi.org/10.1145/2523813

Gasparin, A., Lukovic, S., & Alippi, C. (2022). Deep learning for time series forecasting: The electric load case. *CAAI Transactions on Intelligence Technology*, *7*(1), 1–25. https://doi.org/10.1049/cit2.12060

Gnedenko, B. V. (2018). *Theory of probability*. https://doi.org/10.1201/9780203718964

Hamilton, J. D. (2020). *Time series analysis.* https://doi.org/10.2307/j.ctv14jx6sm

Killick, R., Fearnhead, P., & Eckley, I. A. (2012). Optimal Detection of Changepoints With a Linear Computational Cost. *Journal of the American Statistical Association*, *107*(500), 1590–1598. https://doi.org/10.1080/01621459.2012.737745

Kim, T., Kim, J., Tae, Y., Park, C., Choi, J.-H., & Choo, J. (2022). Reversible instance normalization for accurate time-series forecasting against distribution shift. *International Conference on Learning Representations*, 1–17.

Kingphai, K., & Moshfeghi, Y. (2023). On Time Series Cross-Validation for Deep Learning Classification Model of Mental Workload Levels Based on EEG Signals. *Machine Learning, Optimization, and Data Science*, *13811*, 402–416. https://doi.org/10.1007/978-3-031-25891-6_30

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI'95)*, 1137–1143.

Lainder, A. D., & Wolfinger, R. D. (2022). Forecasting with gradient boosted trees: augmentation, tuning, and cross-validation strategies. *International Journal of Forecasting*, *38*(4), 1426–1433. https://doi.org/10.1016/j.ijforecast.2021.12.003

Li, Y., Salimi-Khorshidi, G., Rao, S., Canoy, D., Hassaine, A., Lukasiewicz, T., Rahimi, K., & Mamouei, M. (2022). Validation of risk prediction models applied to longitudinal electronic health record data for the prediction of major cardiovascular events in the presence of data shifts. *European Heart Journal-Digital Health*, *3*(4), 535–547. https://doi.org/10.1093/ehjdh/ztac061

Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., & Zhang, G. (2018). Learning under Concept Drift: A Review. *IEEE Transactions on Knowledge and Data Engineering*, *31*(12), 2346–2363. https://doi.org/10.1109/tkde.2018.2876857

Page, E. S. (1954). Continuous inspection schemes. *Biometrika*, *41*(1–2), 100–115. https://doi.org/10.1093/biomet/41.1-2.100

Rapach, D. E., & Zhou, G. (2020). *Time-series and cross-sectional stock return forecasting: New machine learning methods*.

Samarajeewa, C., De Silva, D., Manic, M., Mills, N., Moraliyage, H., Alahakoon, D., & Jennings, A. (2024). An artificial intelligence framework for explainable drift detection in energy forecasting. *Energy and AI*, *17*, 100403. https://doi.org/10.1016/j.egyai.2024.100403

Sherstinsky, A. (2020). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network. *Physica D: Nonlinear Phenomena*, *404*, 132306. https://doi.org/10.1016/j.physd.2019.132306

Sugiyama, M., Krauledat, M., & Muller, K.-R. (2007). Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, *8*, 985–1005.

Taleblou, R. (2025). Comparing the performance of different deep learning architectures for time series forecasting. *Journal of Mathematics and Modeling in Finance*, *5*(1), 63–87.

Tsymbal, A. (2004). *The problem of concept drift: Definitions and related work*.

Vollmer, M. A. C., Glampson, B., Mellan, T., Mishra, S., Mercuri, L., Costello, C., Klaber, R., Cooke, G., Flaxman, S., & Bhatt, S. (2021). A unified machine learning approach to time series forecasting applied to demand at emergency departments. *BMC Emergency Medicine*, *21*(1), 9. https://doi.org/10.1186/s12873-020-00395-y

Xiang, Q., Zi, L., Cong, X., & Wang, Y. (2023). Concept Drift Adaptation Methods under the Deep Learning Framework: A Literature Review. *Applied Sciences*, *13*(11), 6515. https://doi.org/10.3390/app13116515

Yuan, L., Li, H., Xia, B., Gao, C., Liu, M., Yuan, W., & You, X. (2022). Recent Advances in Concept Drift Adaptation Methods for Deep Learning. *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, 5654–5661. https://doi.org/10.24963/ijcai.2022/788

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021). Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(12), 11106–11115. https://doi.org/10.1609/aaai.v35i12.17325