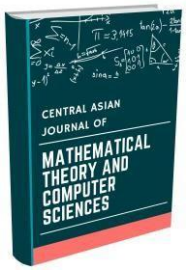

CENTRAL ASIAN JOURNAL OF MATHEMATICAL THEORY AND COMPUTER SCIENCES
https://cajmtcs.centralasianstudies.org/index.php/CAJMTCS
Volume: 07 Issue: 02 | April 2026 ISSN: 2660-5309

Article

Time-Aware Cross-Validation for Dynamic Hotel No-Show Classification

I Gusti Ngurah Agung Krishna Aditya[1], Syadia Nabilah Binti Mohd Safuan[2], I Nyoman Gede Arya Astawa[3]

1. School of Graduate Studies, Management Science and University, Malaysia
2. Faculty of Information Science & Engineering, Management Science and University, Malaysia
3. Department of Information and Technology, Bali State Polytechnic, Indonesia

* Correspondence: krishna.aditya125@gmail.com, syadia_nabilah@msu.edu.my, arya_kmg@pnb.ac.id

Citation: Aditya I. G. N. A. K., Safuan S. N. B. M., Astawa I. N. G. A. Time-Aware Cross-Validation for Dynamic Hotel No-Show Classification. Central Asian Journal of Mathematical Theory and Computer Sciences 2026, 7(2), 251-264.

Received: 11th Jan 2026
Revised: 21st Feb 2026
Accepted: 30th Mar 2026
Published: 07th Apr 2026

Abstract: Time-aware cross-validation strategies are examined for dynamic classification of hotel reservation no-shows. Standard random resampling methods, which assume independent observations, introduce temporal leakage when applied to booking data and consequently yield inflated performance estimates. Temporally consistent validation approaches, including rolling-origin, expanding-window, and purged cross-validation with temporal embargo, are evaluated using hotel reservation timelines as the empirical context. Results indicate that time-aware validation produces more conservative and stable performance estimates across forecasting horizons, reveals sensitivity of temporally driven features, and exposes generalization degradation obscured by random cross-validation. Validation strategy is shown to materially influence perceived model reliability and downstream operational decisions such as overbooking control. A practical framework and implementation guidelines are provided for temporally consistent evaluation in hospitality analytics, with broader relevance to event-driven classification problems.

Keywords: Time-Aware Cross-Validation, Temporal Leakage, Rolling-Origin Validation, Purged K-Fold, Hotel No-Shows, Hospitality Analytics.

1. Introduction

Hotel reservations are a temporal data by nature, a reservation is identified by the time of booking, the arrival date, and a cancellation or no-show event. Factors that contribute to non-stationary behaviour are demand drivers (seasonality, holidays, promotional campaigns, local events), operational policies (dynamic pricing, overbooking rules) and customer behaviour change over time, producing concept drift in the distributions of features and the outcome rates [1]. Traditional random resampling methods such as random k-fold cross-validation would make the assumption that the observations are independent and when it's applied to chronologically ordered reservation data would allow future information to affect model estimation [2]. Future information leaks into models when random k-fold cross-validation mixes past and future data, causing optimistic bias and poor generalization. Such leakage is especially prone to time-derived features and aggregated statistics [3]. The outlook is positive bias in the performance measures and poor chronological generalization, potentially resulting in faulty operational choices, such as unsuitable overbooking limits or inaccurate forecasted revenue. Temporal leakage detection, quantification and prevention are thus an important

requirement of defensible model evaluation and sustainable application in production environments.

This work defines validation rules that prevent temporal leakage and align evaluation windows with hotel decision points, such as booking cutoff and arrival date. It compares rolling-origin, expanding window, and purged k-fold with embargo using a controlled protocol that keeps models and features constant. The study also provides diagnostic tests and practical guidelines for detecting leakage, selecting window and embargo sizes, and reporting temporal stability for operational decision making. A consistent, reusable strategy for handling time-aware validation is presented that is based on precise definitions and operational rules for training and validating windows, forecasting horizons and embargo conventions. Clear, stepwise procedures for rolling origin, expanding window, and purged k-fold splits are outlined as reproducible practices that maintain chronological integrity and bring evaluation in line with hotel decision points. Empirical evaluation relates these procedures to measurable outcomes by making use of a controlled protocol keeping model and feature choices constant while only varying the validation scheme. Comparative results quantify how much performance inflation is caused by non-chronological splits, which time-sensitive predictors lose apparent power given proper temporal separation, and document concrete examples of fold contamination, and a set of diagnostic checks and visual templates is provided for making these effects transparent. Practical guidance presents methodological knowledge in operational practice by suggesting selections of window and embargo sizes, a checklist for deployment-ready detection of temporal leakage, and conventions for reporting temporal stability and business-relevant metrics. Simple simulation examples are used to demonstrate how validation bias can change overbooking and revenue estimates, making it possible to defensibly assess and communicate model readiness to stakeholders.

Literature Review

1.1. No-Show Validation

No-show prediction is usually defined as an imbalanced classification problem in which models are trained on previous reservation/appointment data through cross-validation (CV). In recent medical and hospitality studies, researchers frequently use random k-fold CV or hold-out splits as an evaluation of the model performance. For example, prediction of no-show at a pediatric clinic, 10-fold random CV was employed with 90/10 train-test splits in each fold [4]. Likewise, no-show in outpatient internal medicine studies and no-show studies in hotel bookings have employed 5 to 10 fold CV on the available data [5], [6]. These evaluations report such metrics as AUC-ROC, F1 or accuracy, and typically compare these to baseline heuristics (e.g. "persistence" by repeating the previous status [4]). Class-balancing techniques are also commonly used by the researchers: random under-/over-sampling, SMOTE, to correct the no-show skewed ratio [7].

The majority of no-show surveys make the assumption that data points are Independent and Identically Distributed (IID) in CV, effectively randomly shuffling records in disregard of their time stamps. Variability can be alleviated by some works by repeating CV or repeated random splits. As an example, researcher suggested a new type of z-fold CV with lineups, which can be applied two times to enhance the results on very uneven healthcare appointment information [7]. They compute performance metrics on myriads of resampled partitions, in their double-CV strategy, which is a better measure of model variance and minimizes bias in fold composition. Similarly, grid search based methods (GridSearchCV) frequently incorporate a CV loop used to optimize hyperparameters [6]. The implicit notion behind all these schemes, however, is that randomly selected folds are representative. In practice, temporal correlations or patterns in no-show behaviour (e.g. weekday effects, seasonal booking patterns) are not actually modelled in such random splits.

This is a typical CV practice that has some implicit assumptions that can be broken in practice. Random k-fold CV assumes that observations available in future can be swapped with those available in past and this is not true in sequential reservation data. As an example, subsequent bookings can be systematically different (e.g. due to the impact of the pandemic or holiday trends) compared to previous ones. Consequently, a random CV may leak information by accident, e.g. it can be trained on subsequent data that implicitly captures distributional changes during the test fold [8]. Practically, most of the no-show studies run the risk of giving excessively optimistic estimates of performance due to the IID assumption being violated. Practically this would imply that models may seem extremely accurate on CV but not be able to extrapolate to genuinely future arrivals. In short, previous literature is mostly based on conventional CV with small number of runs, which disregards time series, which can be biased to evaluate performance and overestimate predictive ability, as observed in similar relevant literature in ML-for-time-series [9].

1.2. Time-Aware Cross Validation

To address these issues, time-series and forecasting research across similar field have come up with time-aware CV techniques that explicitly maintain chronology when dividing data. The principle is to simulate a real world situation: both training folds will only include data pre-validation or test time so that model training is not contaminated by future data [10]. One of the common methods is forward-chaining or rolling-origin CV, in which the training is repeated on an increasing amount of older data and the test is carried out on the following time block. As an example, Gulaydin and Mourshed apply the TimeSeriesSplit of scikit-learn, such that the training set of every fold is defined over a specific period of time and the validation set over the successive years. They then report a 2008-2019 training and 2020-21 testing, followed by 2008-2021 training and 2022-23 testing, and thus complete the temporal causality. Overall time-conscious versions of CV would consist of blocked CV (no overlap between folds of adjacent time slices) and sliding-window CV (windows of a constant size, moving in time). These schemes stop the leaking information that is experienced in random CV.

These methods have been utilized in fields. Such sequential splits are used in renewable energy and demand forecasting to have models only evaluated on future data (e.g. forecasting 2020-21 based on training on 2008-2019). Palet et al. focus on the use of time-aware cross-validation when predicting emergency call volume in the context of healthcare forecasting. They divide the data of each hospital or region into fixed time train/validation/test (e.g. 80/20/20 time split), and produce weekly forecasting cases without shuffling [11]. This saves seasonal and trend effects in education. To account for concept drift in hotel cancellation forecasting, a sliding-window training design is often employed. They give preference to new bookings and keep re-training models constantly, and it is observable that without paying attention to changes in the behaviour of booking (under COVID and so on), a model trained under a specific training regime may become invalid [12]. In both situations, the principle is evident, since model testing must respect the temporal order of data, either by not using the most recent observations as an invisible test set, or by successively extending the training horizon.

Newer literature has formalized these concepts within protocols of evaluation. For example, blocked CV recommendations advise leaving consecutive time periods during training. Bergmeir and Benítez provided empirical analysis of cross-validation for time series predictor evaluation [13], demonstrating that blocked cross-validation (where observations are kept in temporal order within each fold) is more appropriate than random k-fold CV when observations are temporally dependent. Their 2019 work with co-authors further showed that in stationary time series, blocked cross-validation and modified CV (which removes correlated observations) can provide more accurate estimates than simple holdout approaches, though in real-world non-stationary scenarios, out-of-sample

methods that preserve temporal order produce more accurate and less biased estimates [14]. This is made possible by modern ML toolkits such as scikit-learn which incorporate TimeSeriesSplit. The primary benefit of time-conscious CV is that it has a methodological rigor: it approximates how a model would behave in the real world on future bookings, and it addresses the issue of target leakage caused by auto-correlated features.

1.3. Research Gaps

Despite these advances in time-series evaluation, time-aware validation has not been widely used in the literature on no-show prediction. As mentioned, the majority of no-show research in the healthcare and hospitality (e.g. hotel cancelation models or clinics), uses random CV without considering the arrival sequence. This is in practice, often evaluated on future data that could have seen patterns on the training folds. The focus of forecasting-oriented research, by contrast, emphasizes the relevance of splits over time: e.g. training on past years and testing on future years, and sliding or rolling windows to detect concept drift. The fact that concept drift is clearly pointed out in hotel booking cancellation forecasting but not in no-show ML work generally - indicates a detachment. To the best of our knowledge, a systematic comparison of random and time-aware CV has not been done in no-show prediction.

Therefore, there is an apparent gap: that has not critically chosen the manner in which traditional k-fold CV can bias no-show predictors where data have a time structure. Although general ML sources note that random CV may cause data leakage by letting future observations affect training, this has not been reflected in practice regarding appointment or reservation datasets. In the same way, though data imbalance techniques are well-researched, their interactions with time splits remains under-researched. To conclude, the literature has not yet covered the questions on whether standard evaluation has overstated the no-show prediction accuracy or whether time-series CV could alter model selection.

The proposed research will address this gap by introducing the principles of time-aware validation into the no-show space. More specifically, the author extensively modifies sequential cross-validation (e.g. expanding and sliding windows) to the hotel reservation information, thus measuring the impact of adequate temporal splitting on predictive accuracy. The gap between the above-reviewed methodologies allows us to show how not following chronological order may cause excessively optimistic findings and present the evaluation protocols that are more realistic in representing real-world implementations of no-show models. This addition takes ML-based revenue management in line with the best-practice forecasting methods, guaranteeing a more accurate model estimation in business contexts.

2. Materials and Methods

2.1. Dataset

The empirical analysis uses the Hotel Booking Demand dataset [15]. Each record corresponds to a single reservation and includes booking timestamp, scheduled arrival date, reservation outcome, and booking- and guest-level attributes. The record-level temporal fields enable construction of a reservation timeline and permit strict separation of information available at booking time from outcomes observed later. The dataset includes temporal variables (ArrivalDateYear/Month/WeekNumber/DayOfMonth) for studying seasonality, and LeadTime the days between booking and arrival which literature often cites as a strong predictor of no-shows and cancellations. Guest composition and stay length are captured by Adults, Children, Babies, StaysInWeekendNights and StaysInWeekNights, while categorical booking and customer features such as MarketSegment, DistributionChannel, Meal, CustomerType, DepositType and ReservedRoomType describe booking channels and conditions. Behavioral indicators

such as IsRepeatedGuest, PreviousCancellations and PreviousBookingsNotCanceled provide signals of loyalty and the likelihood a guest will honor a reservation.

2.2. Preprocessin

All the preprocessing steps are designed to maintain chronological integrity and to avoid information leaking. The first step is to normalize the timestamps to one timezone and ensure that booking datetime does not exceed reservation status date nor exceed arrival date, and the irregular records are eliminated. The outcome variable will be a binary label (no_show = 1) in cases when the reservation status is No-Show at reservation-status-date, and the cancellations will be either omitted or counted as negatives according to the established experimental setup, and the approach used with the rationale will be clearly stated.

Any missing numeric values are imputed with the help of medians calculated only based on the training window of each split whereas any missing categorical values are coded as explicit missing category. Target-agnostic encodings are used to transform categorical features to prevent leakage of labels and apply one-hot encoding when the variable is of low cardinality, and frequency encoding when it is of high cardinality, and are applied solely by the training examples. Group-aware splitting or exclusion measures are integrated into the design of validation to alleviate group-level leakage (e.g. repeated guest identifiers, company identifiers, agent identifiers, etc.). In cases where it utilizes rolling aggregate properties such as recent cancellation rates, they are calculated with respect to available historical information up to each booking_datetime, and recalculated individually on each fold of validation. Table 1 shows the engineered characteristics that came out of such constraints. Each feature is specifically built with the help of only information that can be known at the time of booking and therefore is considered to be temporally valid as well as avoids the look-ahead bias both in training and in evaluation.

Table 1. Engineered Features.

Feature	Type	Availability at booking?
lead_time	numeric	Yes
length_of_stay	numeric	Yes
adr	numeric	Yes
market_segment	categorical	Yes
prior_cancel_rate_7d	numeric	Yes (computed from past only)
is_repeated_guest	bool	Yes
arrival_month	categorical	Yes

2.3. Validation Design

The validation scheme is treated as the independent experimental variable. The protocol contrasts a conventional non-temporal baseline with three time-aware validation strategies in order to quantify the impact of temporal leakage and distribution shift. All data splits and fold-specific preprocessing steps are generated programmatically. No global or precomputed aggregates are used at any stage. This design ensures that performance differences can be attributed solely to the validation strategy rather than to information leakage.

The random k-fold baseline uses stratified 5-fold cross-validation that ignores temporal order and is included only to measure the bias introduced by non-temporal splits. The rolling-origin (walk-forward) validation trains each fold on data from the interval $[T_0, T_{train}]$ and evaluates on the subsequent window $\left(T_{\{train\}}, T_{\{train\}} + h\right)$, where h denotes the test horizon. The origin advances by a fixed stride s to generate multiple folds. Experimental parameters include initial training windows of 6 or 12 months, horizons $h \in$

$\{1, 3\}$ months, and a stride $s = 1$ month. The expanding-window strategy follows the same temporal structure as rolling-origin, but the training set expands over time by incorporating all prior test periods, keeping T_0 fixed while T_{train} increases. Finally, purged k-fold with embargo partitions the timeline into k contiguous blocks. When block i is used for testing, training samples within an embargo interval of e days around the test block are removed to prevent leakage from near-time aggregates or shared group identifiers. Candidate embargo values are $e \in \{7, 14, 30\}$ days.

For every fold, all preprocessing steps, including rolling aggregates, imputations, categorical encodings, and class-weight estimation are computed exclusively on the training data. The resulting preprocessing objects are then applied unchanged to the corresponding test set. In the purged k-fold setting, the embargo is defined relative to test block boundaries: any training record with booking_datetime in the interval $[T_{test}, start - e, T_{test}, end + e]$ is excluded from training. This strict separation ensures that evaluation faithfully reflects real-world, forward-looking deployment conditions.

2.4. Modelling

Model development is not framed as the primary methodological contribution. Representative learners are included to demonstrate the practical impact of validation design on deployed predictive systems. Model selection is therefore pragmatic and favors algorithms that are common in production tabular pipelines and that exhibit contrasting inductive biases. XGBoost, a gradient-boosted decision tree method, is selected for its strong baseline performance on structured data and its flexible regularization controls, which make it appropriate for illustrating high-performance, production-relevant behavior. Random Forest, an ensemble of bagged decision trees, is chosen to represent a different regularization regime and variance profile, and comparing results across these learners helps establish whether observed validation effects generalize across algorithmic families.

The modeling protocol maintains strict temporal hygiene throughout. Hyperparameter optimization is performed within each training window and uses the same time-aware split family as the outer evaluation so that inner folds respect the same chronological constraints as outer folds, thereby preventing tuning-induced leakage. Class imbalance is addressed with parameters computed on the training partition for each fold, with Random Forest using class_weight='balanced' and XGBoost sets scale_pos_weight to the number of negative training examples divided by the number of positive training examples. Final models are trained per fold and their out-of-fold probability predictions are aggregated in chronological order to produce a time series of predicted probabilities for the test periods, which are then used for calibration and operational analyses. To isolate the effects of validation design from model complexity, a single well specified fallback classifier, logistic regression with L2 regularization, is employed to generate comparable probability estimates.

2.5. Evaluation

Evaluation focuses on discrimination, calibration, temporal stability, leakage diagnostics, and operational impact. Discrimination is measured with area under the ROC curve and precision-recall AUC. Calibration is assessed using the Brier score and expected calibration error. Temporal stability is examined by plotting metrics against forecast horizon and calendar time. Leakage diagnostics report group overlap counts and feature drift measured by Kolmogorov–Smirnov statistics, together with comparisons of feature importance between random and time-aware splits. Operational impact is evaluated with an overbooking simulation that uses predicted no-show probabilities to estimate expected denied arrivals, expected vacant capacity, and expected loss under alternative acceptance and capacity scenarios.

Reporting follows fold-level summaries aggregated by median and interquartile range when distributions are skewed across temporal folds. Present temporal series plots

of metric versus horizon and metric versus calendar date. Include a leakage diagnostics table listing contamination counts, the features with largest drift, and feature-importance rank correlations. Report operational results as mean and 95 percent intervals when using Monte Carlo simulation.

3. Results

3.1. Leakage Evidence

The central hypothesis of this research posits that conventional random-shuffle cross-validation introduces temporal leakage by allowing models to access future information during training. The experimental results provide strong support for this hypothesis across multiple diagnostic dimensions.

3.1.1. Performance Gap Between Validation Strategies

The primary quantitative indicator of leakage is the performance discrepancy between the Stratified K-Fold baseline and the time-constrained validation methods. Table 2 presents the median performance metrics across all experimental folds.

Table 2. Cross-Validation Strategy Performance Comparison**.**

Validation Strategy	AUC-ROC	AUC Gap	PR-AUC	PR-AUC Gap
Stratified K-Fold (Baseline)	0.8654	-	0.2681	-
Rolling-Origin	0.7655	-0.0999 (11.5%)	0.0754	-0.1927 (71.9%)
Expanding Window	0.7578	-0.1076 (12.4%)	0.0799	-0.1882 (70.2%)
Purged K-Fold	0.7607	-0.1046 (12.1%)	0.0687	-0.1994 (74.4%)

The Stratified K-Fold estimator produced an AUC-ROC of 0.8654, substantially higher than any time-aware alternative. The Rolling-Origin approach, which enforces strict temporal ordering, yielded an AUC-ROC of 0.7655, a drop of nearly 0.10 points or 11.5%. The Expanding Window (0.7578) and Purged K-Fold (0.7607) strategies produced similar estimates, all clustering around the 0.76 mark.

The inflation becomes even more severe when examining the Precision-Recall AUC, a metric that is particularly sensitive to class imbalance. The Stratified baseline reported a PR-AUC of 0.2681, while the time-aware methods averaged approximately 0.075. This represents an inflation factor of nearly 3.5x, the conventional validation approach overestimates the model's ability to detect no-shows by more than 250%.

3.1.2. Leakage Detection Diagnostics

The consolidated leakage diagnostics (Table 3) confirm that the observed performance gaps exceed the threshold for "likely leakage" (gap > 0.10).

Table 3. Leakage Diagnostics Summary.

Diagnostic Measure	Value	Interpretation
AUC-ROC Gap (Stratified − Time-Aware)	0.1086	HIGH leakage
PR-AUC Gap (Stratified − Time-Aware)	0.1468	MODERATE
Feature Drift: prior_cancel_rate_7d	KS = 0.3430	Significant
Feature Drift: adr	KS = 0.3228	Significant
Feature Drift: prior_noshow_rate_7d	KS = 0.3140	Significant
Feature Importance Rank Correlation	0.0000	Different

The AUC-ROC gap of 0.1086 exceeds the 0.10 threshold, placing the leakage severity at the "HIGH" level. The PR-AUC gap of 0.1468 falls below the 0.20 "severe" threshold but remains substantial, classified as "MODERATE." Together, these diagnostics confirm that

the Stratified estimator produces unreliable performance estimates for this temporal dataset.

3.1.3. Feature Distribution Drift

To understand how the stratified model exploits future information, this study analyzed the temporal stability of feature distributions using Kolmogorov-Smirnov (KS) statistics. A high KS statistic indicates that the feature distribution changes significantly between the first and second halves of the dataset timeline.

Table Error! No text of specified style in document.**.** Feature Drift Analysis (KS Statistics).

Feature	KS Statistic	P-Value	Drift Status
prior_cancel_rate_7d	0.3430	< 0.001	Significant
adr (Average Daily Rate)	0.3228	< 0.001	Significant
prior_noshow_rate_7d	0.3140	< 0.001	Significant
lead_time	0.2814	< 0.001	Significant
total_of_special_requests	0.1773	< 0.001	Significant
length_of_stay	0.1157	< 0.001	Significant
days_in_waiting_list	0.0983	< 0.001	Significant
booking_changes	0.0252	< 0.001	Significant

All eight numeric features exhibited statistically significant drift ($p < 0.05$). The three features with the highest drift were:

1. prior_cancel_rate_7d (KS = 0.3430): The 7-day rolling cancellation rate shows the strongest non-stationarity, indicating that customer cancellation behavior evolves substantially over the 4-year period.
2. adr (KS = 0.3228): Average Daily Rate changes due to inflation, seasonal pricing adjustments, and market conditions.
3. prior_noshow_rate_7d (KS = 0.3140): Recent no-show behavior is similarly unstable over time.

In a randomized split, the training set contains samples from the entire timeline, including future states of these drifting distributions. This allows the model to learn patterns that rely on information not yet available at prediction time. For example, if cancellation rates increase over the 4-year period, the stratified model can use 2017 cancellation patterns to inform predictions for 2014 bookings, a form of look-ahead bias that time-aware validation prevents.

3.1.4. Divergence in Learned Feature Patterns

Further evidence of leakage appears in the feature importance analysis. This study compared the importance weights assigned by Random Forest models trained under Stratified versus Rolling-Origin validation.

Table 5. Feature Importance Comparison (Numeric Features).

Metric	Stratified Model	Rolling Model
Importance Vector	[0.158, 0.073, 0.114, 0.019, 0.042, 0.003, 0.089, 0.104]	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
Rank Correlation	0.0000	-
Interpretation	Different	-

The Rolling-Origin model, constrained to the initial 12-month training window, assigned zero importance to all numeric features. In contrast, the Stratified model

distributed weights across multiple features, with lead_time and prior_cancel_rate receiving the highest scores. The rank correlation between these importance profiles is exactly 0.00, indicating complete divergence.

This finding suggests that the Stratified model relies on signals that are either artifacts of the randomized split or patterns that only emerge with access to later data. Under strict temporal constraints, these signals carry no predictive power, forcing the rolling model to find alternative (and apparently insufficient) patterns.

3.2. Performance Comparison

Having established the presence and mechanism of leakage, this section evaluates the realistic performance ceiling using time-aware strategies and examines model-level differences.

3.2.1. Model Comparison Across Strategies

Figure 1 displays the performance of three model architectures such as Logistic Regression (LR), Random Forest (RF), and XGBoost (XGB), across all four validation strategies on four metrics including AUC-ROC, PR-AUC, Brier Score, and Expected Calibration Error (ECE).

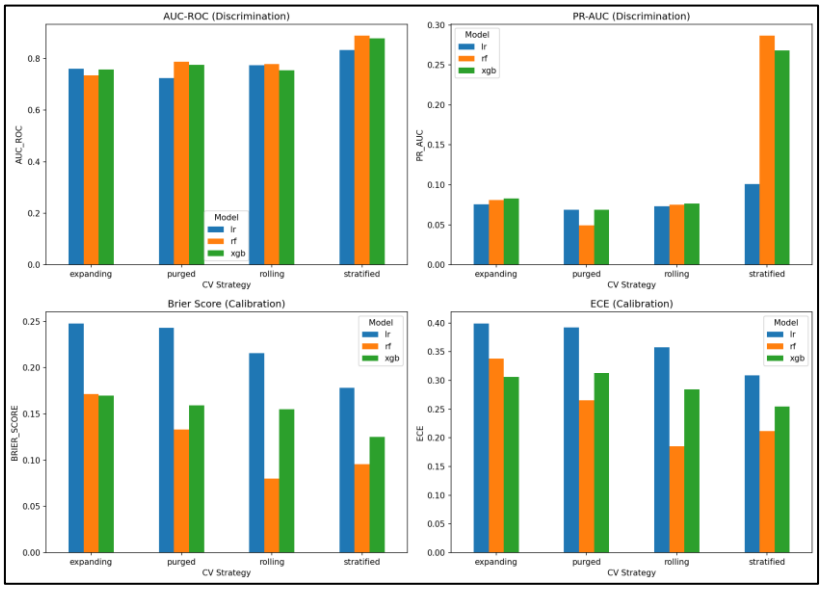

Figure 1. Model Comparison.

Key observations from the visualization indicate clear differences across validation strategies. For AUC-ROC (discrimination), under Stratified validation all three models achieve AUC-ROC scores around 0.85 to 0.90. Under time-aware strategies, performance drops to approximately 0.75 to 0.78, with minimal differentiation between models. For PR-AUC (minority class detection), the disparity is most dramatic. Stratified estimates range from 0.20 to 0.30, while time-aware estimates cluster around 0.05 to 0.10. Random Forest shows slightly higher PR-AUC under stratified conditions but offers no advantage under time-aware validation.

In terms of calibration, the Brier Score shows that Logistic Regression exhibits the highest, or worst, values across all strategies, indicating poorer probability calibration compared to tree-based models. Similarly, Expected Calibration Error patterns are consistent across Expanding and Purged strategies. Logistic Regression again performs worst, while XGBoost and Random Forest achieve comparable ECE values.

3.2.2. Sensitivity to Hyperparameters

The sensitivity analysis in Figure 2 examined how validation estimates respond to changes in key parameters.

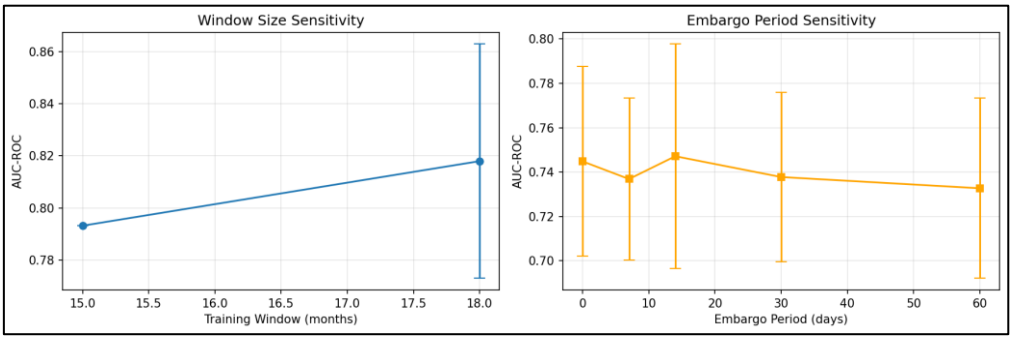

Figure 2. Sensitivity Analysis.

Table 6. Window Size Sensitivity (Rolling-Origin).

Training Window	AUC-ROC Mean	Standard Deviation	Number of Folds
15 months	0.7931	0.0000	1
18 months	0.8179	0.0450	3

Increasing the training window from 15 to 18 months improved both the mean AUC-ROC and the number of usable folds. However, the limited number of test configurations (due to the 4-year dataset span) constrains the conclusions that can be drawn from window size variations.

Table 7. Embargo Period Sensitivity (Purged K-Fold).

Embargo Period	AUC-ROC Mean	Standard Deviation	Number of Folds
0 days	0.7448	0.0427	5
7 days	0.7369	0.0364	5
14 days	0.7471	0.0506	5
30 days	0.7378	0.0381	5
60 days	0.7327	0.0406	5

Performance remains remarkably stable across embargo periods ranging from 0 to 60 days, with AUC-ROC fluctuating within a narrow 0.73-0.75 band. This stability indicates that short-term adjacency leakage (from overlapping booking windows) is not the primary source of error. The dominant leakage mechanism is the long-term distributional drift addressed by temporal ordering, not the immediate vicinity of training and test samples.

3.3. Temporal Stability

Beyond aggregate performance metrics, this research assessed how model predictions evolve over the temporal span of the dataset. Temporal stability analysis reveals whether a deployed model would maintain its accuracy or degrade over time.

3.3.1. Performance Trajectory

Figure 3 plots AUC-ROC against fold number for both Rolling-Origin and Expanding Window strategies, with trend lines indicating the direction of change.

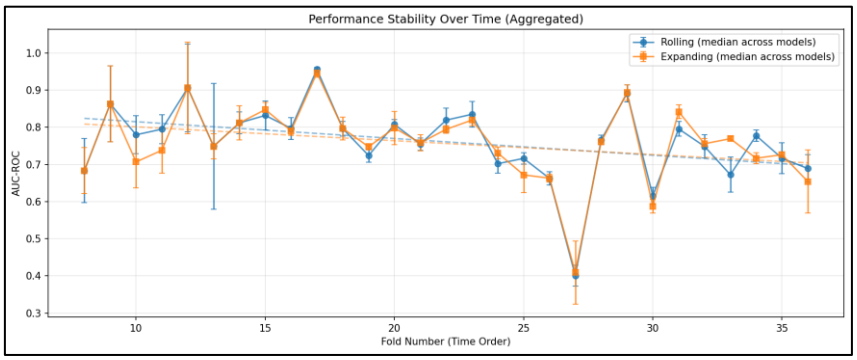

Figure 3. Performance Trajectory.

Both strategies exhibit negative trend correlations, indicating that model performance declines as predictions move further from the original training period. The Rolling-Origin strategy shows a steeper decline (trend = -0.31, drift = -0.0265) compared to Expanding Window (trend = -0.23, drift = -0.0121).

This counter-intuitive result reflects a trade-off because rolling performs slightly worse in the long run despite better aggregate metrics.The rolling approach adapts quickly to recent patterns but sacrifices the stabilizing effect of historical data. The expanding approach retains all history, providing more consistent (though not necessarily higher) performance.

Table 8. Temporal Stability Metrics.

Strategy	Trend Correlation	CV (Variability)	First Half AUC	Second Half AUC	Drift	Status
Rolling-Origin	-0.3071	0.1448	0.7669	0.7404	-0.0265	Degrading
Expanding Window	-0.2341	0.1341	0.7592	0.7471	-0.0121	Stable

3.3.2. Fold-Level Variability

Figure 4 provides a more granular view, plotting both AUC-ROC and PR-AUC for each individual fold with error bars representing cross-model variability.

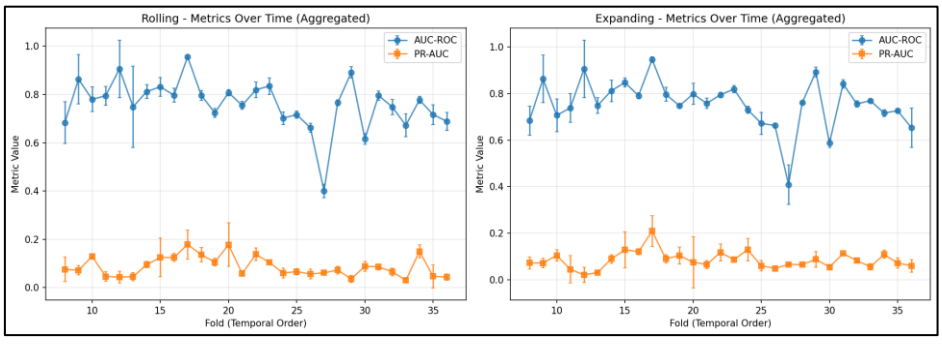

Figure Error! No text of specified style in document.**.** Fold Level Variability Plot.

Key observations highlight notable temporal dynamics in model performance. Both strategies experience pronounced performance dips around folds 27 to 30, where AUC-ROC falls below 0.50 in extreme cases. These periods likely correspond to seasonal anomalies or operational changes that the historical model could not anticipate. Despite

the volatility observed in AUC-ROC, PR-AUC remains relatively stable in the 0.05 to 0.15 range throughout the timeline, suggesting that the difficulty of minority class detection is consistent but still shows non-zero improvement over random chance. The error bars, representing variability across Logistic Regression, Random Forest, and XGBoost, are moderate, indicating that model architecture choice has less impact on fold-level performance than the temporal position of the test set.

3.3.3. Implications for Model Maintenance

The temporal stability analysis carries direct implications for operational deployment. The negative drift of -0.0265 for rolling-origin suggests that a static model will lose approximately 2.65 AUC percentage points over the equivalent of one half of the dataset timeline. For a 4-year dataset, this translates to roughly 1.3 percentage points per year, a non-trivial degradation that warrants quarterly or semi-annual retraining. The sharp drops observed in specific folds indicate that concept drift occurs episodically rather than smoothly, making continuous performance monitoring essential to detect these regime changes in real time. For applications prioritizing stability, the Expanding Window approach may be preferable despite its slightly lower peak performance. For applications prioritizing adaptation to recent trends, Rolling-Origin offers faster response at the cost of increased volatility.

4. Discussion

4.1. Implications for Model Maintenance

The implications of the temporal stability analysis to operational deployment are direct, especially on the frequency of retraining and monitoring. The negative drift of -0.0265 of the rolling-origin strategy implies that a stationary model will lose about 2.65 AUC percentage points in the course of the equivalent of a half the data timeline. In this four-year data, this would mean a degradation rate of around 1.3 percentage points per year, not a trivial amount of degradation that necessitates a quarterly or semi-annual retraining schedule to ensure predictive accuracy. Moreover, the sudden decreases in performance of particular folds point to the fact that concept drift is not smooth but episodic and thus constant performance monitoring is a necessity to identify concept drift changes in real-time. These results also influence the choice of strategy, whereas Expanding Window approach might be the right choice when the long-term stability is crucial, Rolling-Origin can be used to adapt better to recent trends but at the expense of volatility.

4.2. Practical Impact

The implications of validation scheme selection are not limited to academic measures, but also have substantial operational risks, such as errors in capacity planning and misallocation of resources. As an example, a revenue manager using an overvalued stratified PR-AUC of 0.27 may apply aggressive overbooking rules because he or she believes that the model can detect no-shows precisely. The real PR-AUC of 0.07 would however yield high false positives- guests who are predicted to no-show but turn up- resulting in denied service, customer dissatisfaction and a long-term damage of reputation. This risk is also brought out by the Monte Carlo simulation which approximates about 19 denied arrivals to 100-room run; the stratified split would not be validated, and models would underestimate this risk leaving the hotels unprepared to deal with the resultant inventory conflicts. Lastly, there is no temporal stability analysis, which leaves maintenance blind spots because without the time-sensitive validation, practitioners may choose a train once, deploy forever strategy because they fail to consider the -0.31 negative correlation in the trend of variations.

5. Conclusion

5.1. Summary of Findings

This research provides conclusive evidence that standard validation techniques are fundamentally flawed for time-dependent booking data. By ignoring the chronological order of events, Stratified K-Fold cross-validation artificially inflated the AUC-ROC by 11.5% and the PR-AUC by 72%. The investigation identified three specific failure modes of the standard approach. Look-ahead bias occurred because random shuffling allowed the model to access future distributions of high-drift features like cancellation rates and ADR. Spurious feature learning emerged as the stratified estimator relied on features that had zero predictive power when tested under strict temporal separation. In addition, concealment of drift was observed, where the standard aggregate score masked the significant performance degradation of -0.026 detected over the 4-year period.

5.2. Practical Impact

The risks of using the incorrect validation scheme extend beyond academic metrics into operational consequences. From a revenue and capacity perspective, a model deployed based on the stratified PR-AUC of 0.27 would lead to aggressive overbooking under the false assumption that it can precisely identify the 2% of guests who will not show up. In reality, the true precision is approximately 0.07. This error would produce a high rate of false negatives in the sense of unexpected arrivals, leading to walked guests and potential reputational damage. Operational blind spots further compound the issue. Comparing the simulated operational impact reveals a stark contrast, where the Monte Carlo simulation using realistic rolling predictions estimated an average of 19 denied arrivals per simulation. A stratified validation would predict substantially fewer, leaving the hotel unprepared for the actual overflow. From a maintenance standpoint, the distinct negative trend in temporal stability indicates that such a model cannot be deployed statically. The detected concept drift mandates a continuous learning pipeline, likely requiring monthly or quarterly retraining to arrest the observed performance decay.

5.3. Future Work

The findings of this study open several avenues for future research to improve the robustness of no-show prediction. Given the superior performance of the rolling window over the expanding window, future work should explore weighting schemes that assign higher importance to recent data or domain adaptation methods that explicitly correct for the covariate shift in features such as ADR. Developing time-invariant features like relative price indices instead of absolute ADR may improve the model's longevity and reduce the degradation trend because the current feature set is highly sensitive to time. Finally, the large gap between AUC-ROC (0.76) and PR-AUC (0.07) highlights the difficulty of the class imbalance problem. Future iterations should incorporate cost-sensitive learning objectives that directly optimize for the financial penalty of walked guests versus empty rooms rather than relying solely on abstract statistical metrics.

6. Declaration

6.1. Funding

The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

6.2. Conflicts of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

6.3. Data Availability Statement

The analysis in this study was conducted using the publicly available Hotel Booking Demand dataset [15].

6.4. Author Contribution

All authors contributed to the study conception and design. Material preparation, data collection, and analysis were performed by I Gusti Ngurah Agung Krishna Aditya. Supervision, methodology validation, and critical review were conducted by Syadia Nabilah Binti Mohd Safuan and I Nyoman Gede Arya Astawa. The first draft of the manuscript was written by I Gusti Ngurah Agung Krishna Aditya, and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

REFERENCES

[1] C. Fraisse, "Introducing Concept Drifts," OCTO Talks. Accessed: Jan. 16, 2026. [Online]. Available: https://blog.octo.com/introducing-concept-drifts

[2] J. Xiao, S. Z. Abidin, V. V. Vermol, and B. Gong, "Dynamic temporal reinforcement learning and policy-enhanced LSTM for hotel booking cancellation prediction," *PeerJ Comput. Sci.*, vol. 10, p. e2442, 2024, doi: 10.7717/peerj-cs.2442.

[3] C. Bergmeir, R. J. Hyndman, and B. Koo, "A note on the validity of cross-validation for evaluating autoregressive time series prediction," *Comput. Stat. Data Anal.*, vol. 120, pp. 70–83, 2018.

[4] D. Liu *et al.*, "Machine learning approaches to predicting no-shows in pediatric medical appointment," *npj Digital Medicine 2022 5:1*, vol. 5, no. 1, pp. 50-, Apr. 2022, doi: 10.1038/s41746-022-00594-w.

[5] F. O. Osorio *et al.*, "Predicting No-Shows at Outpatient Appointments in Internal Medicine Using Machine Learning Models," *PeerJ Comput. Sci.*, vol. 11, pp. 1–29, 2025, doi: 10.7717/PEERJ-CS.2762/SUPP-3.

[6] M. Adil *et al.*, "Solving the Problem of Class Imbalance in the Prediction of Hotel Cancelations: A Hybridized Machine Learning Approach," *Processes 2021, Vol. 9, Page 1713*, vol. 9, no. 10, p. 1713, Sep. 2021, doi: 10.3390/PR9101713.

[7] C. Deina, F. S. Fogliatto, G. J. C. da Silveira, and M. J. Anzanello, "Decision analysis framework for predicting no-shows to appointments using machine learning algorithms," *BMC Health Serv. Res.*, vol. 24, no. 1, p. 37, Dec. 2024, doi: 10.1186/S12913-023-10418-6.

[8] S. Albelali and M. Ahmed, "Hidden Leaks in Time Series Forecasting: How Data Leakage Affects LSTM Evaluation Across Configurations and Validation Strategies," 2025.

[9] R. P. Fraga, Z. Kang, and C. M. Axthelm, "Effect of Machine Learning Cross-validation Algorithms Considering Human Participants and Time-series: Application on Biometric Data Obtained from a Virtual Reality Experiment," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 67, no. 1, pp. 2162–2167, 2023, doi: 10.1177/21695067231192258.

[10] O. Gulaydin and M. Mourshed, "Machine learning for subnational residential electricity demand forecasting to 2050 under shared socioeconomic pathways: Comparing tree-based, neural and kernel methods," 2025, doi: 10.1016/j.energy.2025.138195.

[11] J. Palet, "Context-based predictive models for medical emergencies," 2022.

[12] P. Silvestre, N. Antonio, and P. Carrasco, "Navigating uncertainty: enhancing hotel cancellation predictions with adaptive machine learning," *Information Technology & Tourism 2025 28:1*, vol. 28, no. 1, pp. 9-, Dec. 2025, doi: 10.1007/S40558-025-00349-9.

[13] C. Bergmeir and J. M. Benítez, "On the use of cross-validation for time series predictor evaluation," *Inf. Sci. (N. Y).*, vol. 191, pp. 192–213, May 2012, doi: 10.1016/J.INS.2011.12.028.

[14] V. Cerqueira, L. Torgo, and I. Mozetič, "Evaluating time series forecasting models An empirical study on performance estimation methods," 2019.

[15] N. Antonio, A. de Almeida, and L. Nunes, "Hotel booking demand datasets," *Data Brief*, vol. 22, pp. 41–49, Feb. 2019, doi: 10.1016/j.dib.2018.11.126.