ADIA Lab Structural Break Open Benchmark Challenge

Furui Wang
Shanghai Business School

Feishu Institute

January 29, 2025

Competition Overview

- **ADIA Lab Structural Break Open Benchmark Challenge**
- Organized by **ADIA Lab** and hosted on the **CrunchDAO Hub**
- The challenge focuses on detecting **structural breaks** in univariate time series under diverse and noisy data-generating processes.
- Participants are provided with:
  - Time series with a **candidate breakpoint**.
  - A binary label indicating whether the breakpoint is real.
- The goal is to design robust models that generalize across heterogeneous series and structural change patterns.

Problem Motivation

- **Structural break** refers to abrupt changes in the underlying data-generating process of a time series.
- In real-world applications such as finance, economics, and shipping markets, structural breaks often indicate regime shifts, policy changes, or external shocks.
- Accurate detection of structural breaks is crucial for:
  - Improving forecasting performance.
  - Enhancing risk management.
  - Supporting timely decision-making.
- However, traditional statistical methods may struggle under noisy data, small sample sizes, or complex distributional changes.

Dataset Description

- Data from **ADIA Lab Structural Break Open Benchmark** (CrunchDAO)
- Synthetic **univariate time series** with a **given boundary point**
- Train: **10,000** series Test: **100**
- Length per series: **1,000–5,000** observations
- Format: `DataFrame (id, time)` with `value`, `period`

Data Description

id	time	value	period
0	0	-0.005564	0
	1	0.003705	0
	2	0.013164	0
	3	0.007151	0
	4	-0.009979	0
...	...	...	...
10000	2134	0.001137	1
	2135	0.003526	1
	2136	0.000687	1
	2137	0.001640	1
	2138	0.001074	1

23715734 rows × 2 columns

Figure: X train: each `id` corresponds to a time series with a given boundary separating pre- and post-boundary segments.

id	structural_breakpoint
0	False
1	False
2	True
3	False
4	False
...	...
9996	False
9997	False
9998	False
9999	False
10000	True

10001 rows × 1 columns

Figure: y train: Ground-truth labels indicating whether the given boundary is a true structural break.

Task Definition and Visual Motivation

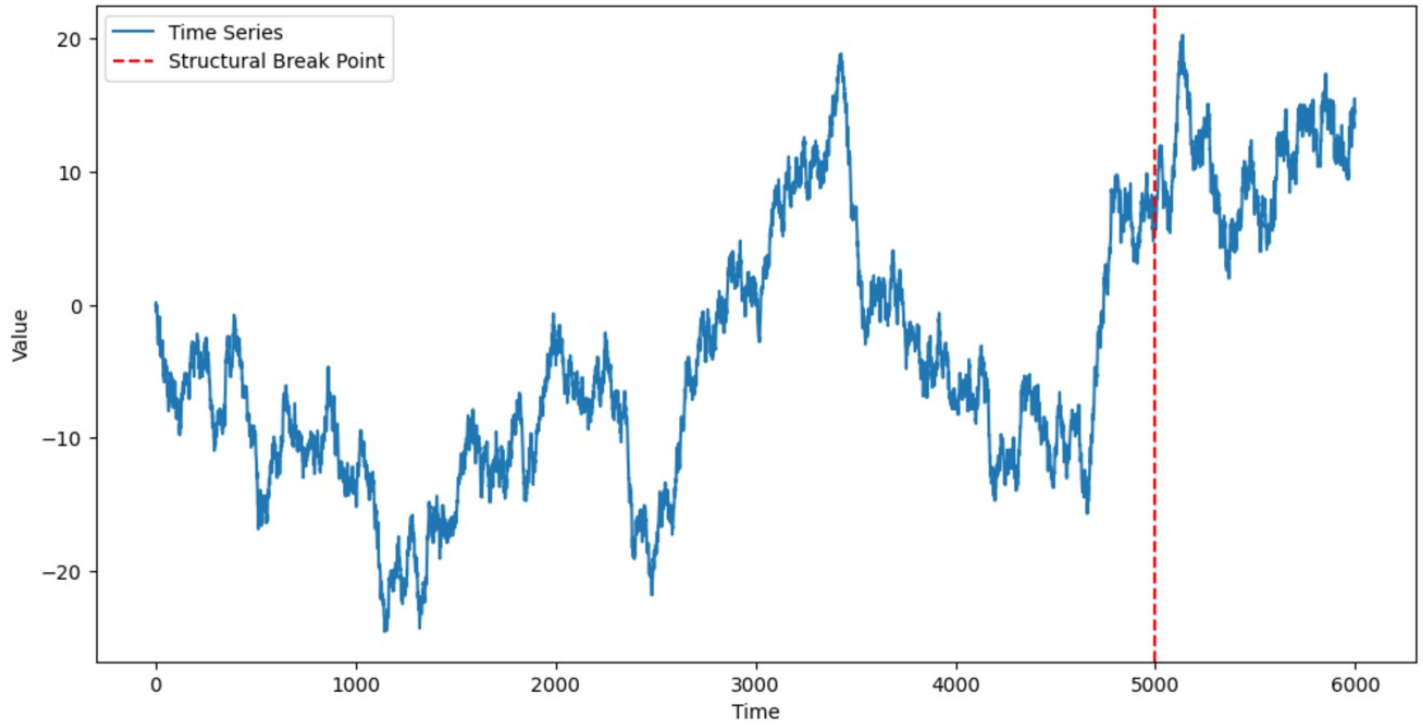

Figure: Raw univariate time series with an underlying structural break.

- Raw time series are often visually ambiguous under noise.
- Structural breaks may be difficult to identify from the original scale.

Task Definition and Visual Motivation

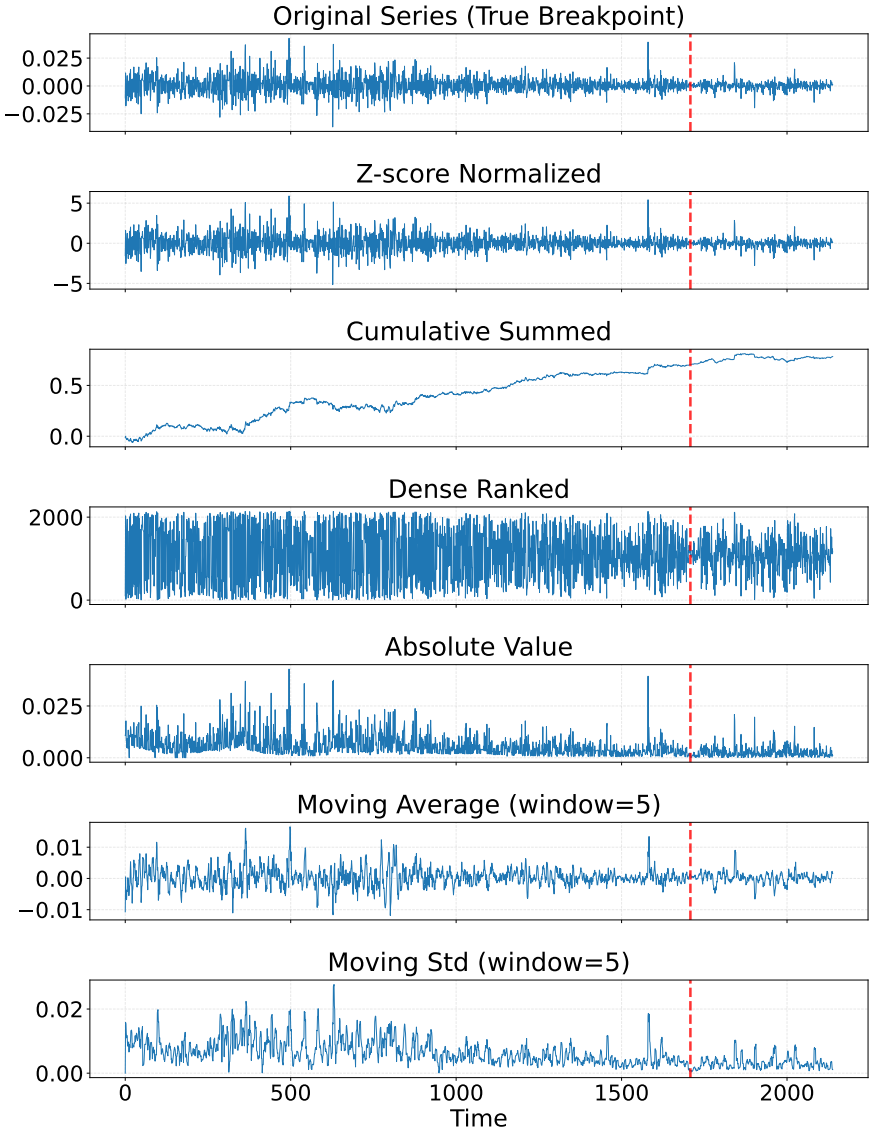

Figure: True break transformations.

- Simple deterministic transformations enhance breakpoint visibility.
- The same structural break becomes more distinguishable across transformed views.
- These transformations reveal the break more clearly than the raw series.
- The visual distinction is crucial for accurate break detection.

Modeling Process

- **Step 1:** Evaluate individual statistical tests.
- **Step 2:** Construct deterministic feature representations capturing global, temporal, and pre/post-boundary changes.
- **Step 3:** Compare standard learning models under strict out-of-fold evaluation.
- **Step 4:** Enhance boosting models by incorporating TabPFN probability outputs as additional features.
- **Step 5:** Build a stacked model that integrates TabPFN-informed features with boosting for improved generalization.

Performance of Individual Statistical Tests

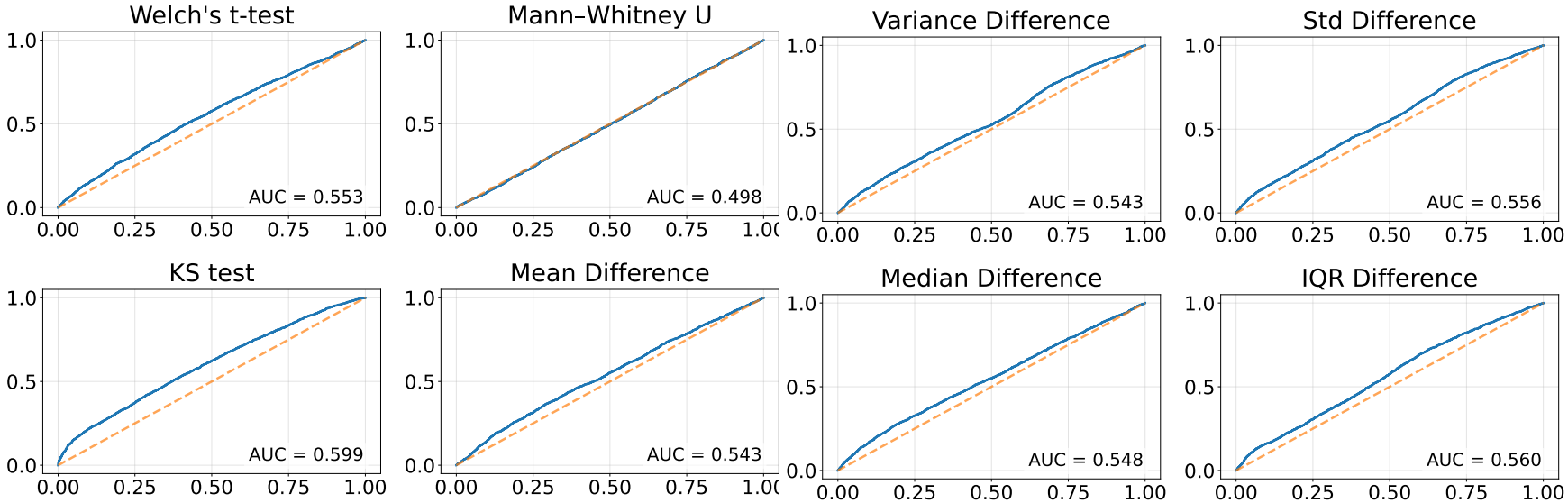

Figure: ROC curves of individual hypothesis-test statistics.

- Some hypothesis-test statistics exhibit only weak or inconsistent discriminative power when used in isolation.
- While insufficient as standalone detectors, these statistics provide complementary signals that can be integrated as informative features.

Feature Construction

We construct a set of **deterministic and interpretable features** to characterize structural differences around the candidate breakpoint.

- **Global statistics** Mean, standard deviation, skewness, and normalized location statistics summarizing the overall distribution.
- **Temporal-shape features** Rolling mean and variance, rank dynamics, and z-score normalization capturing local temporal patterns.
- **Pre/Post comparison features** Differences and ratios of mean, variance, IQR, and MAD between segments before and after the boundary.
- **Statistical test features** Test statistics from F-test, Levene test, KS test, and Welch's t-test quantifying distributional shifts.

In total, we obtain a compact yet expressive feature representation for each time series instance.

Model-Level Comparison (Method Perspective)

Table: Model-level comparison of classification methods from a methodological perspective.

Model	Method Essence	Key Strengths	Main Limitations
Logistic Regression	Linear probabilistic classification model	Highly interpretable, stable baseline, fast training	Limited to linear decision boundaries
SVM (RBF Kernel)	Margin-based nonlinear kernel method	Strong performance in small-sample and high-dimensional settings	Sensitive to kernel choice and hyperparameter tuning
Random Forest	Bagging-based ensemble of decision trees	Robust to noise, low overfitting risk, little tuning required	Weak at capturing complex global structures
Gradient Boosting	Sequential boosting of weak decision trees	High flexibility, captures nonlinear feature interactions	Prone to overfitting without careful regularization
LightGBM	Histogram-based gradient boosting tree model	Efficient, scalable, strong performance on tabular data	Sensitive to noisy features and parameter settings
XGBoost	Regularized gradient boosting framework	Strong generalization, handles feature interactions well	Computationally expensive, tuning-sensitive
CatBoost	Ordered boosting with categorical feature handling	Robust to overfitting, minimal preprocessing needed	Slower training and higher memory usage

Model Performance on Training Set (Strict OOF)

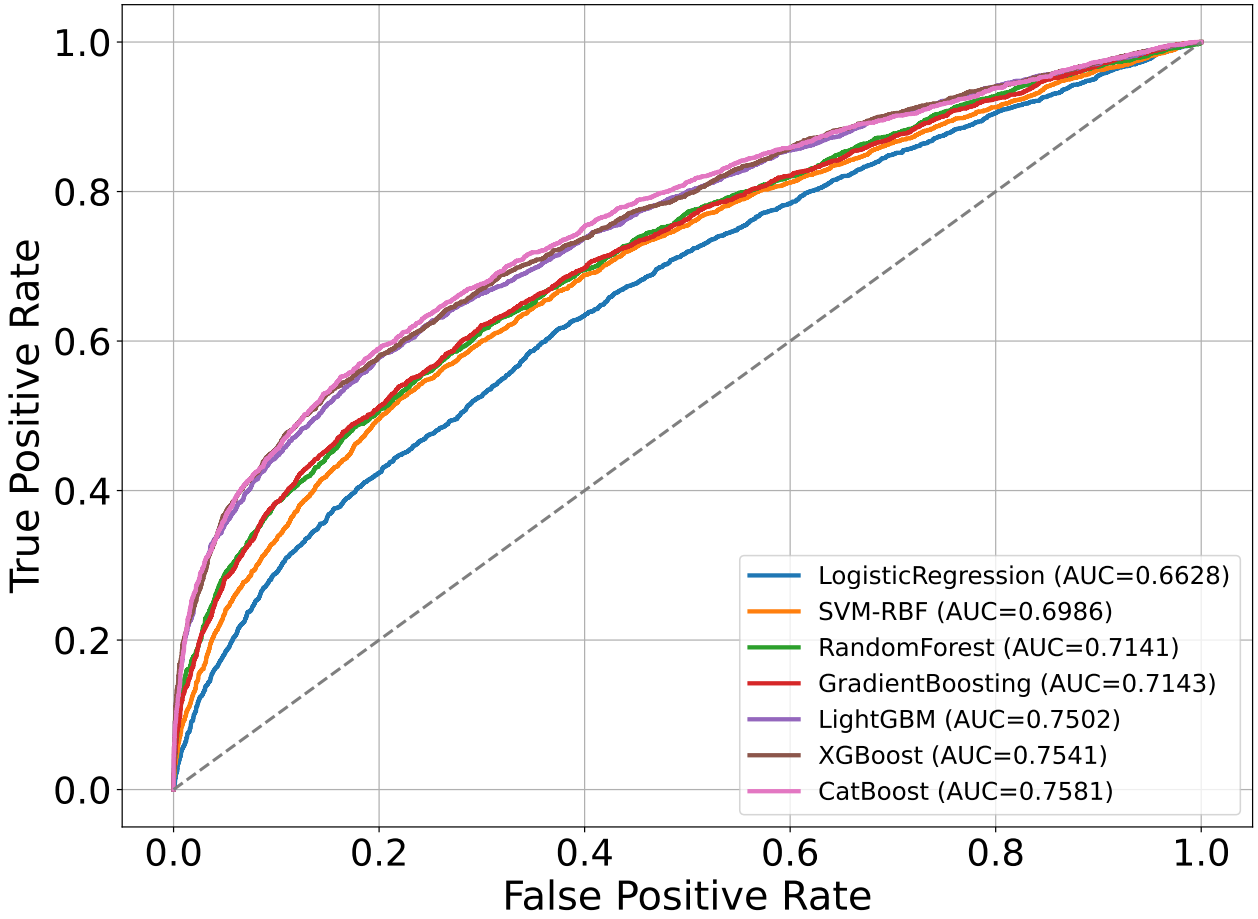

Figure: Under strict OOF evaluation, **LightGBM**, **XGBoost**, and **CatBoost** outperform other models, demonstrating the effectiveness of boosting-based ensembles for structural break detection.

Model Performance on Test Set

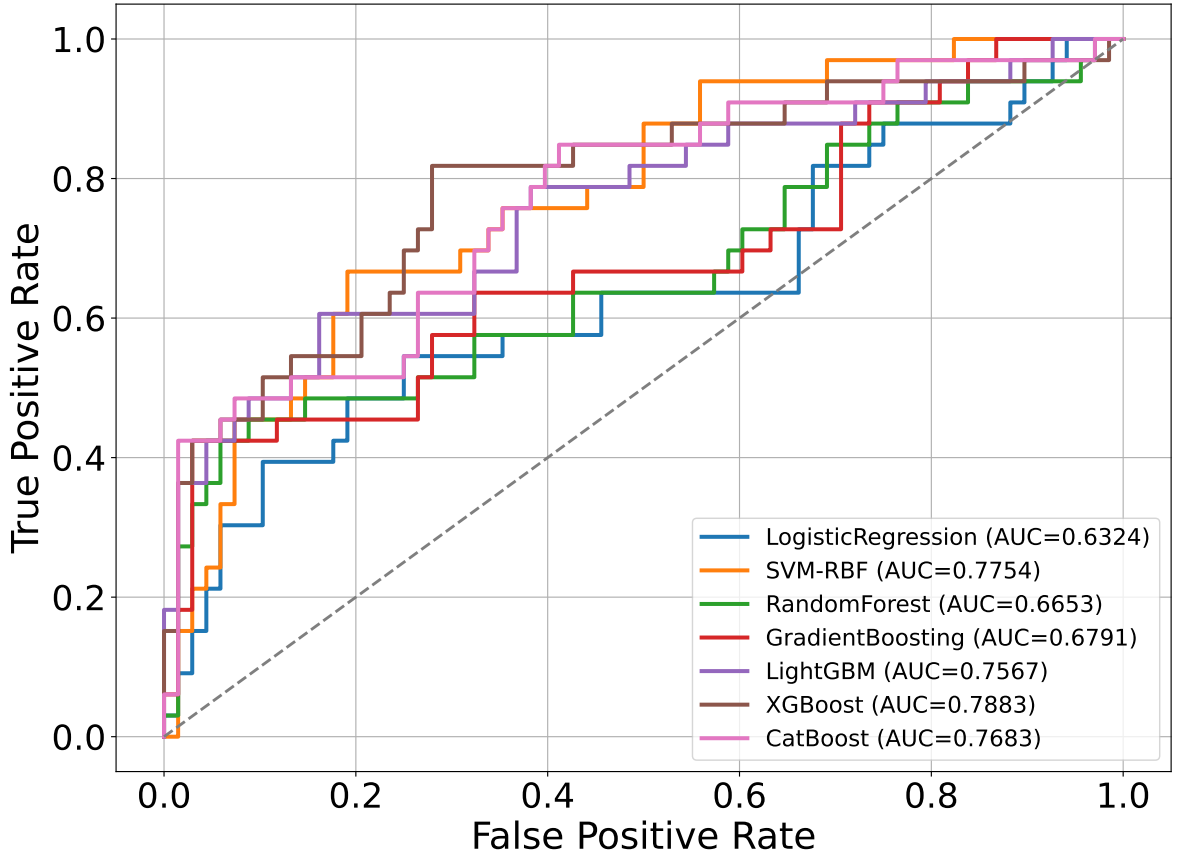

Figure: **LightGBM**, **XGBoost**, and **CatBoost** consistently achieve strong and stable performance under strict OOF and test-set evaluation, while SVM, despite good test-set results, shows weaker and less stable OOF behavior and is therefore excluded from the final model selection.

TabPFN as a Prior-informed Feature Generator

- TabPFN is a **pre-trained tabular model** that performs conditional inference without task-specific training.
- It encodes strong **prior knowledge** over generic tabular data distributions.
- We apply TabPFN **only to global statistical features**, which provide a compact and stable summary of each time series.
- This design avoids high-dimensional and noisy inputs, while enabling TabPFN to produce a reliable **OOF probability feature**.
- The TabPFN output is treated as an additional feature and combined with structural and temporal features in boosting models.

Boosting Models with TabPFN-Augmented Features(Training)

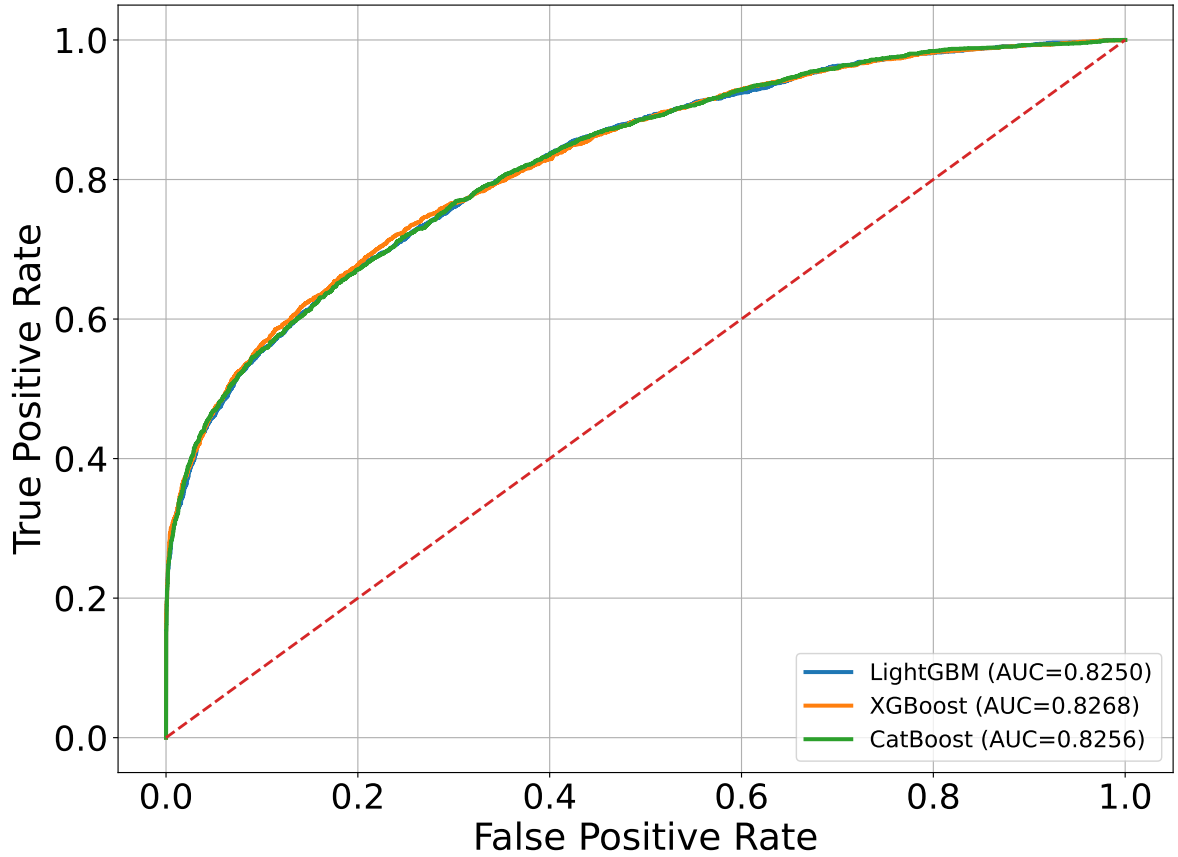

Figure: TabPFN is trained on global statistical features to generate an OOF probability feature, which is concatenated with the remaining feature groups and used to train LightGBM, XGBoost, and CatBoost, leading to strong and stable OOF performance.

Boosting Models with TabPFN-Augmented Features(Test)

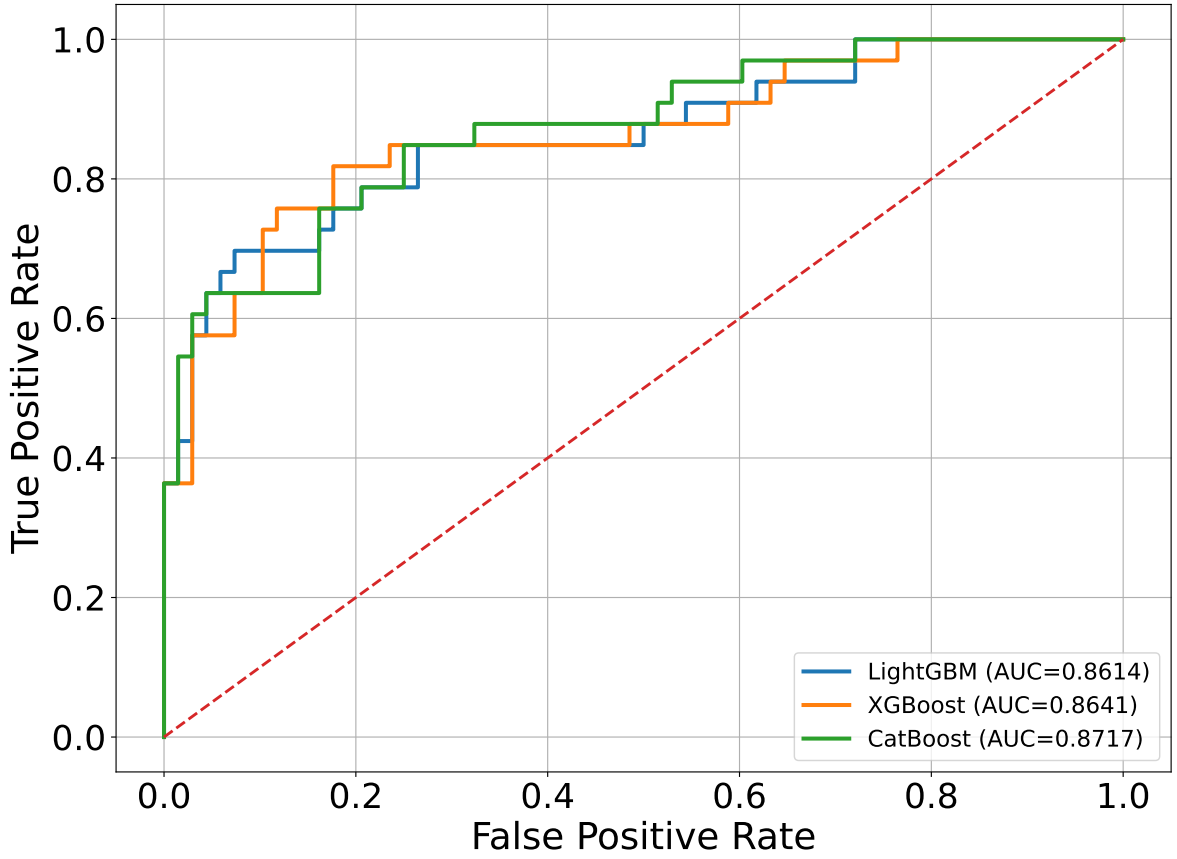

Figure: Augmenting boosting models with the TabPFN-derived probability feature consistently improves generalization on the independent test set.

Final Stacked Model: TabPFN + CatBoost

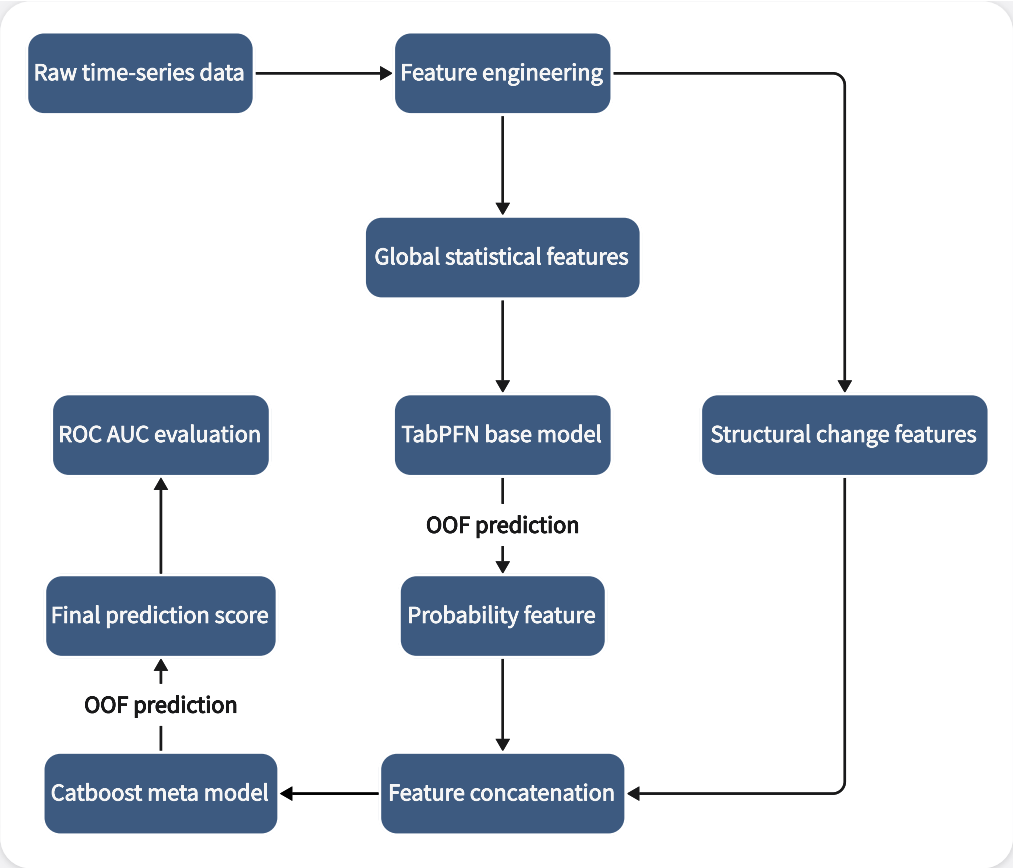

Figure: Stacked Model Pipeline.

- **Motivation:** Combine TabPFN's small-sample robustness with boosting's nonlinear capacity.
- **Stage 1 — TabPFN:** Trained on global statistical features (X_1), generating leakage-free OOF probabilities.
- **Stage 2 — CatBoost:** OOF probabilities are concatenated with structural change features (X_S) for final prediction.

Final Stacked Model: Leakage-Free Performance

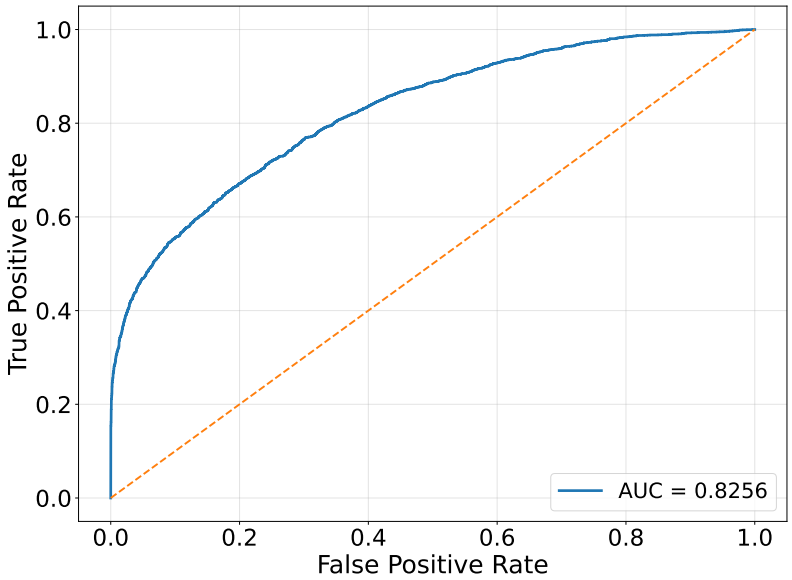

Figure: Training ROC, AUC ≈ 0.8256.

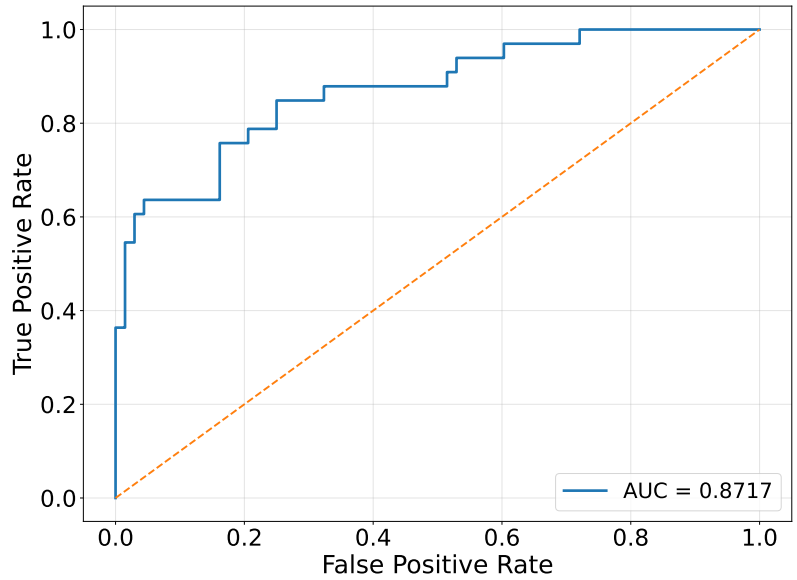

Figure: Test ROC, AUC ≈ 0.8717.

- **Leakage-free evaluation:** TabPFN generates strict out-of-fold (OOF) probabilities, which are used as meta-features for **CatBoost**.
- **Consistent generalization:** Comparable and improved AUC on the independent test set indicates strong robustness under distribution shift.

Feature Interpretability

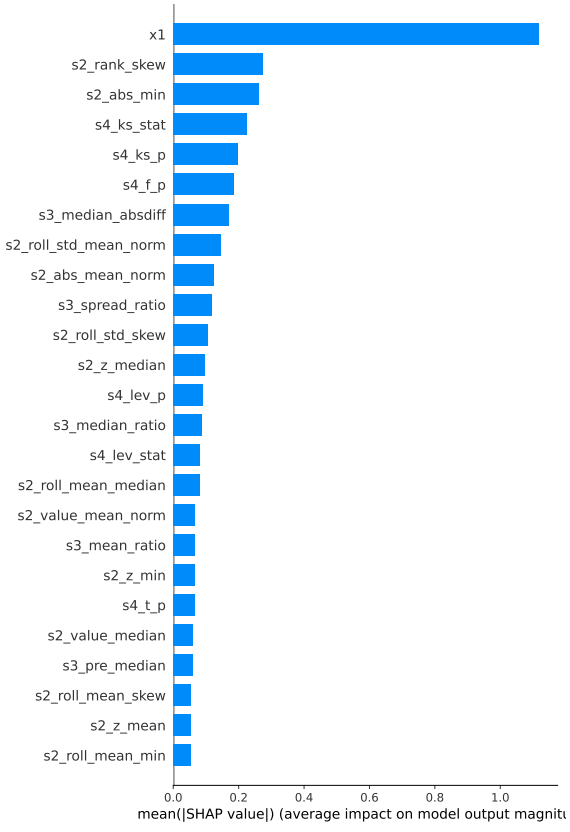

Figure: Global importance.

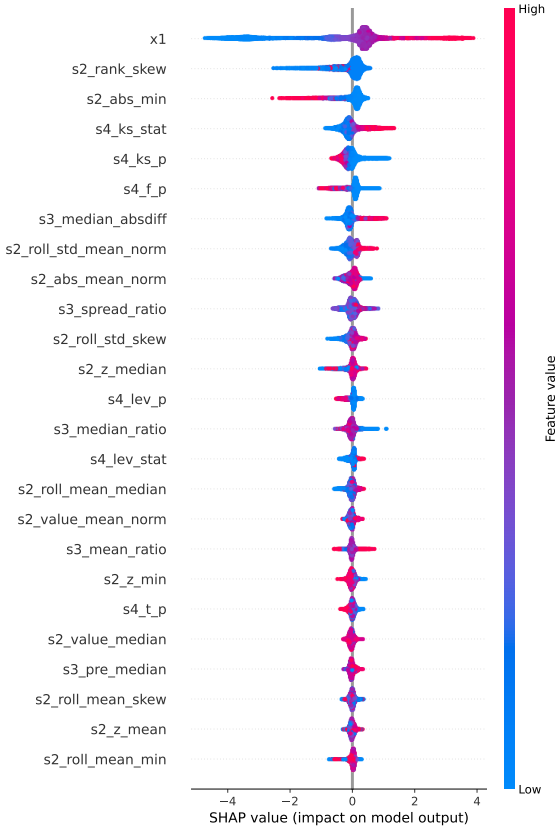

Figure: Local effects.

- `x1` (TabPFN base score) and structural features contribute most to the final decision.
- Beeswarm patterns show consistent directional effects, supporting stable feature usage.

Distributional Separation of High-Importance Features

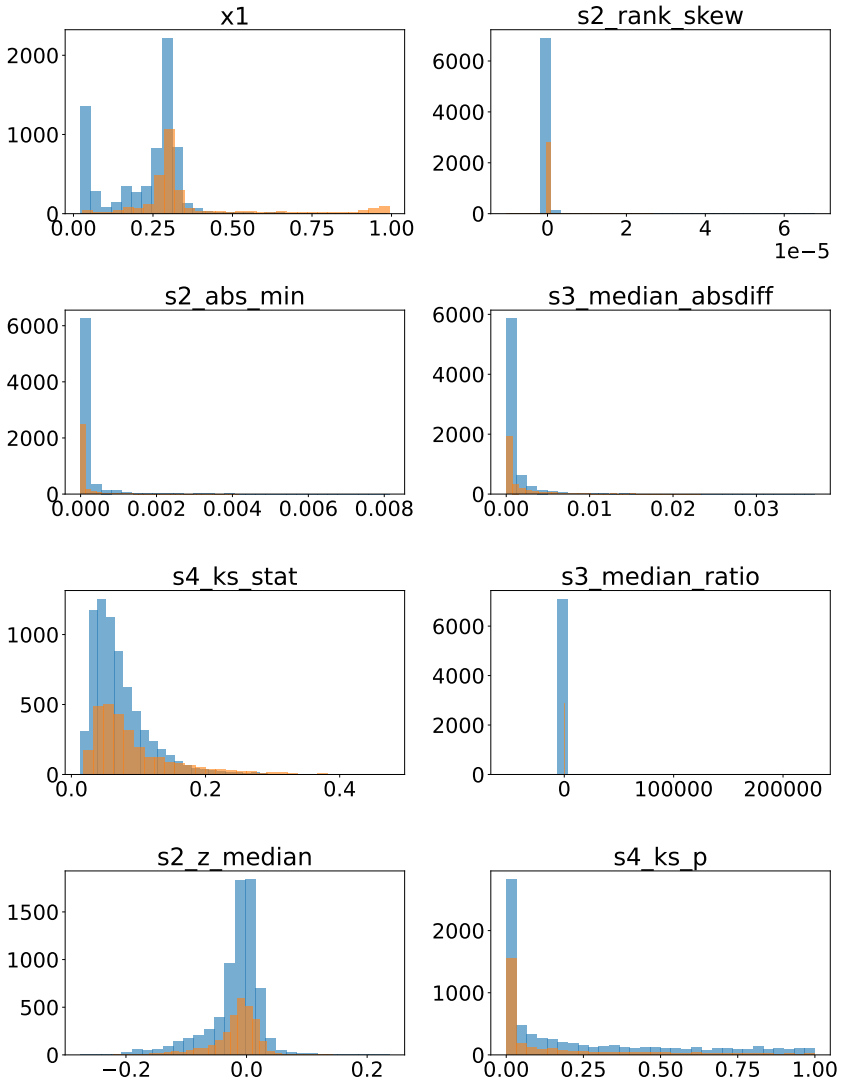

Figure: Top-8 feature distributions.

- High-importance features show clear distributional shifts across classes.
- Temporal-shape features exhibit heavier tails or increased dispersion under breaks.
- Statistical-test features concentrate in distinct value ranges.
- Results confirm genuine data-level separability prior to modeling.

Conclusion

- We propose a deterministic and model-agnostic feature framework for structural break detection at a known boundary.
- Individual statistical tests are weak in isolation, while learning-based models provide clear performance gains.
- TabPFN shows strong generalization under strict out-of-fold evaluation, and **CatBoost** effectively captures nonlinear feature interactions.
- A two-stage stacked model (**TabPFN + CatBoost**) achieves robust, leakage-free, and consistent performance on both training and test sets.

Principled features + leakage-free evaluation + complementary models.

Thank you!