Leakage-Safe Benchmark Design for Market-Stress Early Warning: An Economically Credible Evaluation

Ting Liu *

Posted Date: 25 March 2026

doi: 10.20944/preprints202603.1925.v1

Keywords: market stress; early warning systems; leakage-safe evaluation; reproducible benchmarking; machine learning in finance

 doi:10.20944/preprints202603.1925.v1

Article

Leakage-Safe Benchmark Design for Market-Stress Early Warning: An Economically Credible Evaluation

Ting Liu

McCormick school of engineering, Northwestern University, 2145 Sheridan, Evanston, IL 60208, USA; tingliu2027@u.northwestern.edu

Abstract

This study develops a leakage-safe benchmark design for market-stress early warning and examines whether conclusions about model usefulness remain stable under temporally ordered, economically credible evaluation. The analysis compares alternative models across feature sets, forecast horizons, drawdown thresholds, walk-forward schemes, and threshold-selection rules within a unified out-of-sample framework. Performance is assessed using standard discrimination measures, including PR-AUC and ROC-AUC, together with operational criteria relevant to monitoring decisions, including false alarms per year, event hit rate, median lead time, and alert-budget diagnostics. The results show that benchmark design materially affects model rankings and that systems appearing strong under conventional statistical metrics are not always preferred once operational tradeoffs are taken into account. These findings suggest that benchmark specification should be treated as part of the empirical question rather than as a neutral background choice. The paper contributes to the computational evaluation of financial early-warning systems by showing how leakage-safe benchmark design shapes model comparison and decision-relevant conclusions.

Keywords: market stress; early warning systems; leakage-safe evaluation; reproducible benchmarking; machine learning in finance

1. Introduction

Financial markets periodically enter episodes of elevated stress, abrupt repricing, and sustained drawdown that matter for both private risk management and broader financial surveillance. Under changing market conditions, such episodes can emerge and develop rapidly, so early-warning systems are widely used to identify periods in which short-horizon downside risk becomes unusually high. Yet the practical value of such systems cannot be inferred from predictive accuracy alone. A warning rule that performs well on conventional statistical metrics may still be unattractive in use if it generates excessive false alarms, provides insufficient lead time, or exceeds the monitoring capacity of decision-makers. Market-stress early warning is therefore not only a forecasting problem but also a problem of economic evaluation, because model usefulness depends on alert burden, asymmetric costs, and the operational environment in which warnings are acted upon.

A broad literature studies financial distress prediction, volatility forecasting, drawdown monitoring, and machine-learning-based risk classification [2,3,8,16]. However, much of this work evaluates candidate systems under a relatively fixed benchmark environment. Feature sets, forecast horizons, event definitions, walk-forward schemes, and decision thresholds are often treated as secondary implementation choices rather than as substantive determinants of empirical conclusions. In temporally ordered financial applications, that treatment is too restrictive. A system that appears attractive under one horizon, event definition, or threshold rule may be materially less useful under another, and a benchmark that emphasizes severe events, rapid adaptation, or low false-alarm tolerance may favor a different model altogether. Benchmark design is therefore not merely a technical detail. It is part of the

 doi:10.20944/preprints202603.1925.v1

economic comparison problem because it shapes which monitoring system appears preferable under the decision objective being considered.

A related difficulty is that financial prediction exercises are especially vulnerable to information leakage and weak reproducibility [10–12]. Out-of-sample performance can be overstated when preprocessing, threshold calibration, feature scaling, or event construction uses information that would not have been available in real time. When the evaluation protocol is not fully transparent, reported model differences are also difficult to audit or replicate. These concerns are particularly important in market-stress monitoring, where claims of predictive usefulness are meaningful only if they survive leakage-safe evaluation under a transparent and economically interpretable design.

This study develops a leakage-safe benchmark design for market-stress early warning and uses it to examine whether conclusions about model usefulness remain stable under economically credible evaluation. The analysis compares alternative warning systems across feature sets, forecast horizons, drawdown thresholds, walk-forward schemes, and threshold-selection rules within a unified out-of-sample setting [13]. Performance is assessed not only with conventional discrimination measures, including PR-AUC and ROC-AUC, but also with operational criteria directly relevant to monitoring decisions, including false alarms per year, event hit rate, median lead time, alert-budget diagnostics, and cost-sensitive comparisons. The objective is not simply to report which model ranks first under a single preferred specification, but to study how computational design choices alter the economic interpretation of predictive usefulness.

The paper makes four contributions. First, it shows that benchmark specification should be treated as part of the empirical question and studies how alternative benchmark choices alter both statistical and operational conclusions. Second, it implements a leakage-safe walk-forward protocol that strictly separates model estimation, threshold selection, and evaluation over time. Third, it broadens model assessment by combining standard classification metrics with operational criteria tied directly to economic monitoring usefulness. Fourth, it provides a transparent empirical structure that makes benchmark results easier to audit, interpret, and extend in leakage-sensitive financial applications.

The main empirical finding is that model rankings are materially sensitive to benchmark design. Across benchmark specifications, the relative standing of candidate systems changes with the feature set, forecast horizon, drawdown threshold, walk-forward design, and threshold rule. In addition, conclusions suggested by conventional discrimination metrics do not always coincide with those implied by operational monitoring criteria. The broader implication is that economically credible comparison in market-stress early warning requires more than strong predictive scores alone. It requires leakage-safe evaluation, operationally meaningful performance criteria, and explicit treatment of the benchmark choices that govern model comparison in practice.

The remainder of the paper is organized as follows. Section 2 reviews related work on market-stress monitoring, financial prediction benchmarks, and leakage-aware evaluation in time-series settings. Section 3 presents the data and benchmark design. Section 4 presents the leakage-safe evaluation protocol. Section 5 describes the candidate models, and Section 6 introduces the performance measures. Section 7 reports the empirical findings. Section 8 describes the reproducible pipeline. Section 9 discusses the implications and limitations, and Section 10 concludes.

2. Related Work

Research on market-stress early warning draws on several overlapping literatures, including financial distress prediction, macro-financial early-warning systems, volatility and correlation modeling, and more recent machine-learning approaches to financial forecasting. The closest connection here is to work that treats risk monitoring as an out-of-sample prediction problem rather than primarily as a question of structural asset-pricing interpretation. Early studies of financial distress and default established the importance of forward-looking risk indicators and showed that predictive performance depends heavily on variable construction and evaluation design. Structural approaches such as Merton [2] provided a foundational framework linking balance-sheet fragility to default risk, while later

 doi:10.20944/preprints202603.1925.v1

empirical work showed that reduced-form and market-based variables can improve failure prediction in practice [3]. Although this literature is largely concerned with firm-level distress, it helped establish the broader principle that financial risk monitoring should be assessed against explicit forward targets and under genuine out-of-sample evaluation.

A second relevant strand examines early-warning indicators for systemic stress, banking crises, and broader macro-financial vulnerabilities. This work shows that financial crises are often preceded by measurable buildups in leverage, credit growth, asset-price imbalances, and related vulnerabilities, and that the usefulness of such indicators depends not only on whether they contain predictive signal, but also on how that signal is evaluated in real time [4,6]. Related contributions from central banks and policy institutions likewise emphasize that early-warning systems should be assessed with explicit attention to operational tradeoffs, since missed crises and false alarms impose different costs and have different timing implications [7]. This perspective is closely related to the focus here on event detection, threshold calibration, and monitoring usefulness. At the same time, that literature typically emphasizes macro-financial vulnerability indicators or policy-oriented composite indexes rather than the comparative evaluation of alternative market-stress models under multiple economically meaningful benchmark choices.

A third body of work concerns volatility, correlation, and other market-based measures of financial risk. A large literature shows that volatility and dependence dynamics contain valuable information for forecasting and risk management, especially during episodes of market turbulence [8,9]. In practice, simple volatility-based systems remain important baselines because they are interpretable, operationally transparent, and often difficult to outperform consistently under realistic out-of-sample conditions [18]. For that reason, the empirical design in this study treats a volatility-based approach as a primary benchmark rather than as a minor robustness check. At the same time, market stress may also be reflected in changing cross-asset comovement, dispersion, and nonlinear interactions, making it empirically important to compare flexible machine-learning models against parsimonious statistical benchmarks within a common leakage-safe evaluation design.

The study is also related to the growing literature on machine learning in finance and on the evaluation of predictive systems in time-series settings. A recurring lesson from this literature is that apparent performance gains can be highly sensitive to data partitioning, model selection, and threshold tuning. In finance, concerns about data snooping and backtest overfitting have long been recognized: repeated specification search on the same data can lead to overly optimistic inference, even when each individual exercise appears careful [10,11]. These concerns are directly relevant for market-stress early warning, where conclusions may shift with the feature set, forecast horizon, event definition, or decision threshold. Existing studies often report model quality under a single benchmark design, even though operationally relevant conclusions may depend on whether the monitoring objective is tactical or strategic, whether the user places greater weight on severe drawdowns or broader deterioration, and whether threshold calibration prioritizes alert reliability or broader event coverage. The approach taken here is therefore to treat benchmark design itself as an economically meaningful source of ranking variation rather than as a fixed background choice [16].

More specifically, the paper connects to work on leakage control and credible benchmarking in predictive modeling. In time-series environments, even subtle look-ahead contamination in preprocessing, threshold estimation, or validation design can materially distort out-of-sample conclusions. Benchmark-oriented research in adjacent machine-learning domains has accordingly stressed the importance of standardized evaluation protocols, transparent data processing, and reusable research pipelines for producing credible comparisons [14]. The same concern is particularly acute in financial applications, where temporal ordering is intrinsic to the problem and implementation details can alter measured performance in economically meaningful ways [12]. In this sense, leakage control is not merely a technical hygiene issue; it is necessary for preserving the economic interpretability of out-of-sample evidence. A benchmark that is not leakage-safe can exaggerate the practical usefulness

 doi:10.20944/preprints202603.1925.v1

of a warning system and produce misleading conclusions about the relative merits of alternative models.

Relative to the existing literature, the contribution of this paper lies in the economically credible computational comparison of market-stress warning systems under alternative benchmark designs. Prior research has established the value of early-warning indicators, the relevance of volatility and vulnerability measures, and the importance of out-of-sample validation. The narrower but consequential gap addressed here is that conclusions about model quality are often reported under a single benchmark specification, even though those conclusions may change materially when the feature universe, forecast horizon, drawdown threshold, walk-forward scheme, or threshold-selection rule is altered. These dimensions are economically meaningful because they correspond to different monitoring horizons, different tolerances for false alarms relative to missed stress episodes, and different assumptions about regime persistence versus adaptation. By focusing on leakage-safe evaluation, ranking sensitivity, and operationally relevant model assessment, the paper aims to clarify how benchmark design shapes the substantive conclusions drawn from computational model comparison in financial early-warning applications.

3. Data, Benchmark Design, and Analysis Outputs

3.1. Pipeline Structure and Analysis Inputs

The empirical workflow is organized in two stages. First, an upstream benchmark-construction pipeline prepares the benchmark inputs, generates walk-forward model predictions, and saves the resulting benchmark outputs. Second, the paper-analysis workflow reads these saved outputs directly in `R`, constructs the benchmark panel in memory, computes ranking and stability summaries, and generates the reported figures and tables.

At the paper-analysis stage, the workflow operates on saved benchmark objects that include pooled summaries, alert-budget summaries, robustness summaries, and row-level out-of-sample prediction files. The benchmark-analysis script validates the required fields, restricts the sample to the benchmark scope of the study, constructs specification-level comparison panels, and produces the reported figures directly from code rather than through manual spreadsheet editing.

The benchmark is organized around two feature-set definitions, `cross_asset` and `full_benchmark`. Calendar alignment, feature construction, target construction, and sample trimming are handled in the upstream benchmark pipeline. The paper-analysis layer therefore focuses on transparent aggregation, comparison, and visualization of the saved benchmark outputs.

3.2. Feature Sets and Event Definition

The benchmark evaluates two predictor sets, denoted cross_asset and full_benchmark. Let $\mathbf{x}_t^{(\mathrm{CA})}$ denote the predictor vector under the cross_asset specification, and let $\mathbf{x}_t^{(\mathrm{FB})}$ denote the corresponding predictor vector under the full_benchmark specification.

Feature definitions, transformations, and lookback rules are implemented within the benchmark-construction pipeline. Where predictor scaling is used, normalization is performed within each walk-forward fold using training data only:

$$z_{j,t} = \frac{\tilde{x}_{j,t} - \mu_{j,\mathrm{train}}}{\sigma_{j,\mathrm{train}}},$$

where $\mu_{j,\mathrm{train}}$ and $\sigma_{j,\mathrm{train}}$ are estimated solely from the training observations of the relevant fold.

Let $R_{t,t+H}$ denote the realized forward return over horizon H, and let $D_{t,t+H}$ denote the realized forward drawdown over the same horizon. The benchmark target is a binary market-stress label defined from forward downside outcomes, while predictors and alert rules are constructed using only information available at date t.

 doi:10.20944/preprints202603.1925.v1

When the event rule is threshold-based, the label is defined as

$$y_t = \begin{cases} 1, & \text{if } D_{t,t+H} \leq -dd, \\ 0, & \text{otherwise,} \end{cases}$$

where dd denotes the benchmark drawdown threshold.

This forward-looking event construction is evaluated strictly out of sample within the walk-forward design described below. Any preprocessing or thresholding required for target construction is implemented using training information only within the relevant fold. This training-only construction is a core requirement of the leakage-safe design.

3.3. Benchmark Grid and Model Scope

The benchmark varies five design dimensions: feature set, forecast horizon, drawdown threshold, walk-forward mode, and threshold-selection rule. The feature set takes values cross_asset and full_benchmark; the forecast horizon is $H \in \{20, 63\}$; the drawdown threshold is $dd \in \{0.08, 0.10\}$; the walk-forward mode is either expanding or rolling; and the threshold-selection rule is either precision_target or f1. Together these choices define

$$2 \times 2 \times 2 \times 2 \times 2 = 32$$

benchmark specifications.

The main model comparison focuses on three models: Random Forest, XGBoost, and a Volatility Baseline. Because each main model is evaluated over the full 32-specification grid, the main benchmark panel contains $3 \times 32 = 96$ model-specification evaluations. The appendix comparison expands the model set to six models by additionally including Logit, Correlation Baseline, and Dispersion Baseline, yielding $6 \times 32 = 192$ model-specification evaluations.

Let $\hat{p}_t$ denote the model-implied stress probability. The operational alert rule is defined as

$$a_t = \mathbb{1}\left\{\hat{p}_t \geq \tau^{\text{train}}\right\},$$

where τ^{train} is selected using training data only under the relevant threshold-selection rule.

The headline primary specification used for paper-level comparison is cross_asset with $H = 63$, $dd = 0.08$, rolling windows, and the precision_target threshold rule. All substantive conclusions are then evaluated across the full 32-specification benchmark grid.

3.4. Leakage-Safe Walk-Forward Evaluation

All models are evaluated under a leakage-safe walk-forward design using two windowing schemes: expanding and rolling. In each benchmark specification, the model is estimated on a historical training block and then evaluated on the immediately subsequent out-of-sample test block. Repeating this procedure through time produces the saved walk-forward predictions used in the benchmark summaries.

Under expanding windows, the training sample grows as additional observations become available. Under rolling windows, the training sample advances through time using a moving historical window. In both cases, each test observation occurs strictly after the data used for estimation, preprocessing, threshold calibration, and any fold-specific normalization.

Let $\theta^{(k)}$ denote the fitted parameter vector in fold k. Then

$$\theta^{(k)} = \arg\min_{\theta \in \Theta} \mathcal{L}\left(\mathcal{D}_{\text{train}}^{(k)}; \theta\right),$$

where $\mathcal{D}_{\text{train}}^{(k)}$ denotes the fold-k training data and Θ the admissible parameter set of the implemented model class. No test-period information is used in the corresponding fold.

 doi:10.20944/preprints202603.1925.v1

The benchmark-analysis script aggregates these fold-level outputs into specification-level summaries, a benchmark panel, ranking summaries, robustness summaries, and figure-ready datasets constructed directly in memory within R. This design ensures that model comparison is conducted under a genuinely out-of-sample protocol and that the resulting benchmark summaries are directly traceable to the leakage-safe walk-forward evaluation.

3.5. Evaluation Metrics and Benchmark Summaries

Model performance is evaluated using both statistical and operational criteria. Statistical discrimination is summarized using precision, recall, F1, ROC-AUC, and PR-AUC. Operational usefulness is summarized using false alarms per year, event hit rate, median lead time, and alert-budget diagnostics.

The benchmark-analysis workflow constructs several derived summaries from the saved outputs. First, it builds a main benchmark panel for the three headline models over the 32 benchmark specifications. Second, it computes ranking summaries across specifications, including average ranks for PR-AUC, event hit rate, and false alarms per year, as well as counts of how often each model attains the best value under a given criterion. Third, it computes robustness summaries such as performance ranges and rank standard deviations across specifications. Fourth, it generates appendix-style tables that extend the comparison to the additional baseline models.

This design makes it possible to evaluate not only absolute performance levels, but also whether relative model rankings remain stable when the benchmark design is varied.

3.6. Row-Level Diagnostics and Overlay Availability

Some paper components operate on row-level out-of-sample predictions rather than on specification-level summary tables. These include timeline diagnostics and the stylized defensive-monitoring overlay. The paper-analysis script explicitly checks whether row-level predictions are available for the headline primary specification before producing such diagnostics.

In the current benchmark outputs, row-level predictions are not available for the paper-primary specification

$$(\texttt{cross\_asset},\ H = 63,\ dd = 0.08,\ \texttt{rolling},\ \texttt{precision\_target}),$$

so timeline diagnostics for that headline specification are not produced. When the optional defensive-monitoring overlay is enabled in `auto_available` mode, the script instead uses the nearest available `cross_asset` row-level specification and labels that choice explicitly in the console output and paper draft.

This distinction is important for reproducibility: paper-level benchmark comparisons are always based on the intended primary specification and the full benchmark grid, whereas row-level diagnostic illustrations are only produced for specifications for which the required saved prediction fields are actually available.

3.7. Reproducibility and Executable Workflow

All empirical analyses are implemented in R. The paper-analysis workflow uses standard packages including dplyr, tidyr, ggplot2, readr, scales, stringr, purrr, forcats, and lubridate. The analysis environment is documented through standard dependency records such as sessionInfo() output and/or an renv.lock file.

The empirical workflow is structured to support reproducibility. Benchmark-construction scripts, manuscript-analysis scripts, configuration files, and saved benchmark outputs are organized so that the reported tables and figures can be generated directly from code. The workflow documentation records the asset universe, sample construction, feature definitions, target construction, walk-forward evaluation design, threshold-selection rules, and benchmark-specification settings used in the study.

The paper-analysis script reads benchmark outputs, validates the required columns, constructs the benchmark panel, computes ranking and robustness summaries, and generates all reported figures directly from code. This includes the primary-comparison figure, specification-sensitivity heatmaps,

 doi:10.20944/preprints202603.1925.v1

robustness heatmaps, operational trade-off plots, average-rank summaries, and alert-budget curves. Appendix comparisons are produced from the same workflow by extending the model set to the additional baseline specifications.

Because the contribution of the paper is explicitly framed around leakage-safe benchmarking and reproducible empirical evaluation, all benchmark summaries and figures are generated programmatically from saved outputs rather than through manual post-processing.

Table 1. Empirical design summary

Item	Specification used in this study
Implementation language	`R`
Feature sets	`cross_asset`, `full_benchmark`
Forecast horizons	$H \in \{20, 63\}$
Drawdown thresholds	$dd \in \{0.08, 0.10\}$
Walk-forward modes	`expanding`, `rolling`
Threshold rules	`precision_target`, `f1`
Number of benchmark specifications	32
Main models	Random Forest, XGBoost, Volatility Baseline
Appendix models	Logit, Correlation Baseline, Dispersion Baseline
Main benchmark rows	$3 \times 32 = 96$ model-specification evaluations
Appendix benchmark rows	$6 \times 32 = 192$ model-specification evaluations
Primary specification	`cross_asset`, $H = 63$, $dd = 0.08$, `rolling`, `precision_target`
Primary evaluation metrics	Precision, Recall, F1, ROC-AUC, PR-AUC
Operational metrics	False Alarms/Year, Event Hit Rate, Median Lead Time, alert-budget diagnostics
Benchmark-analysis outputs	Benchmark panel, ranking summary, robustness summary, figure-ready datasets, appendix comparison panel
Row-level diagnostics	Produced only when required row-level predictions are available for the requested specification
Software	`dplyr`, `tidyr`, `ggplot2`, `readr`, `scales`, `stringr`, `purrr`, `forcats`, `lubridate`

Section 7 first reports headline comparisons under the primary specification and then evaluates ranking sensitivity, operational trade-offs, and robustness across the full benchmark grid.

4. Leakage-Safe Protocol

A central objective of this study is to ensure that benchmark performance is evaluated under a genuinely out-of-sample market-stress early-warning setting. In financial time-series prediction, leakage can arise not only from explicit look-ahead bias, but also from subtler implementation choices involving preprocessing, threshold calibration, label handling, and model selection [12]. The protocol adopted here is therefore designed so that forecasting, thresholding, and evaluation are aligned strictly with the information that would have been available at each decision date.

4.1. Temporal separation and walk-forward evaluation

The protocol is built on strict temporal separation between model estimation and evaluation. On each forecast date t, the model is trained only on observations that precede the corresponding test block, and predictions are generated without access to future returns, future labels, or future threshold information. If $\mathcal{I}_t$ denotes the information set available at time t, then all estimation and alert-generation steps must depend only on $\mathcal{I}_t$ and on historical data observed before the forecast target window.

 doi:10.20944/preprints202603.1925.v1

To preserve this chronology, the benchmark uses walk-forward evaluation rather than random sample splitting. For fold k, the procedure takes the form

$$\mathcal{D}_{\text{train}}^{(k)} \rightarrow \mathcal{D}_{\text{test}}^{(k)},$$

and is repeated sequentially through time until the sample is exhausted. Under the expanding mode, the training sample grows as new observations become available. Under the rolling mode, the training sample advances with a fixed-length historical window. In both cases, every test observation occurs strictly after the data used for estimation.

4.2. Leakage-safe label construction and preprocessing

The benchmark target is a forward drawdown event over horizon H. Because the label at time t is defined using future market realizations, it must be treated strictly as an ex post evaluation object. The future drawdown outcome is used only to assess the forecast once the horizon has elapsed and is never allowed to influence real-time predictor construction, model estimation, or alert calibration.

The same principle applies to preprocessing. Any transformation of predictors that depends on estimated quantities, including scaling or other normalization, is treated as part of the model-fitting pipeline and is re-estimated within each walk-forward fold using training data only. Global full-sample preprocessing is not permitted, since it could transmit information from the evaluation period back into the estimation stage.

4.3. Training-only threshold calibration

The benchmark distinguishes between probability prediction and binary alert generation. Each model produces an estimated event probability, but operational use requires a threshold above which an alert is issued. Threshold selection is therefore a potential source of hidden leakage if it is calibrated using the full sample or tuned directly on future test outcomes.

To prevent this, threshold selection is performed using training data only within each walk-forward step. For a given threshold rule, such as `precision_target` or `f1`, the threshold is estimated on the historical training segment and then applied unchanged to the subsequent out-of-sample test observations. This ensures that operational metrics such as false alarms per year and event hit rate retain their intended real-time interpretation.

4.4. Common-protocol model comparison

All competing models are evaluated under the same temporal protocol, event definition, threshold-selection logic, and metric computation within each specification. This common-protocol design ensures that performance differences are attributable to the models or feature sets themselves rather than to inconsistent evaluation mechanics.

The benchmark also avoids ex post adaptation within a given test path. Specifications are defined in advance, and results are reported across an explicit benchmark grid rather than retrospectively optimized on the basis of realized test outcomes. The aim is therefore not to validate one specification through hindsight, but to assess how model rankings change across pre-specified and leakage-safe evaluation choices.

4.5. Summary

The leakage-safe protocol rests on four principles: temporal separation between estimation and evaluation, walk-forward testing that preserves chronology, training-only preprocessing and threshold calibration, and common-protocol comparison across models. These design choices do not guarantee that the highest-ranked model will remain optimal in all future environments, but they do ensure that reported benchmark performance is measured under rules that respect the chronology of the prediction problem.

 doi:10.20944/preprints202603.1925.v1

5. Models

This section describes the model classes included in the benchmark. The purpose is not to introduce a new predictive architecture, but to compare representative models under a common leakage-safe evaluation protocol. The model set is chosen to span both flexible machine-learning methods and simpler market-based baselines, so that predictive gains can be evaluated relative to transparent and practically interpretable alternatives.

The main benchmark compares three primary models: Random Forest, XGBoost, and a volatility baseline. Appendix analyses extend the comparison to logit, correlation-based, and dispersion-based baselines.

5.1. Random Forest

Random Forest is a tree-based ensemble method that aggregates predictions across many decision trees estimated on resampled subsets of the training data. It is well suited to financial classification problems because it can accommodate nonlinear effects and variable interactions without requiring a fully parametric specification. In the present benchmark, Random Forest serves as a representative nonlinear ensemble learner with a strong balance between predictive flexibility and out-of-sample stability [15].

5.2. XGBoost

XGBoost is a gradient-boosted tree ensemble that constructs predictions sequentially by fitting new trees to the residual errors of the existing ensemble. Relative to Random Forest, boosting often provides greater predictive flexibility and may exploit richer feature spaces more aggressively. In this benchmark, XGBoost is included as a higher-capacity nonlinear comparator whose performance can be assessed under the same walk-forward and thresholding framework as the other models [5].

5.3. Volatility baseline

The volatility baseline is included as a parsimonious and interpretable comparator motivated by the central role of volatility in practical market monitoring. In many applications, a volatility-based warning rule is easier to interpret and communicate than a high-capacity machine-learning model [8,9]. Its inclusion therefore provides a direct test of whether added model complexity translates into materially better early-warning performance under a common out-of-sample protocol [1].

5.4. Appendix models

In addition to the three primary models, the appendix extends the benchmark to include a logit specification, a correlation baseline, and a dispersion baseline. The logit model provides a standard parametric benchmark for binary classification, while the correlation and dispersion baselines capture simpler market-structure signals that may still be informative for stress detection. These auxiliary models are not central to the main benchmark narrative, but they help place the primary comparison in a broader context.

5.5. Common evaluation framework

All models are evaluated under the same leakage-safe protocol, using the same event definitions, forecast horizons, walk-forward schemes, threshold rules, and performance metrics within each specification. This common-protocol design is essential because it ensures that performance differences reflect the models or feature sets themselves rather than inconsistent evaluation mechanics.

The model set is intentionally selective rather than exhaustive. Random Forest and XGBoost represent flexible nonlinear learners, the volatility baseline provides a parsimonious market-based comparator, and the appendix models widen the comparison with additional simple alternatives. This is sufficient for the paper's central objective, which is to study benchmark sensitivity under a structured and reproducible evaluation framework rather than to survey every possible forecasting method.

 doi:10.20944/preprints202603.1925.v1

6. Metrics

This section defines the performance metrics used in the benchmark. Because market-stress early warning is both a statistical prediction problem and an operational monitoring problem, performance cannot be summarized adequately by a single measure. The benchmark therefore evaluates models using complementary discrimination and operational metrics.

6.1. Probability forecasts and binary alerts

Each model produces an out-of-sample score or estimated probability of a future market-stress event. Let $\hat{p}_t \in [0,1]$ denote the model output for date t. These continuous outputs are informative because they allow models to be compared as ranking systems. A practical warning framework, however, must also convert them into binary alerts. Let τ denote the decision threshold selected under the leakage-safe training procedure described in Section 4. The corresponding binary warning indicator is

$$\hat{y}_t = \mathbf{1}\{\hat{p}_t \geq \tau\},$$

where $\hat{y}_t = 1$ indicates that the model issues an alert and $\hat{y}_t = 0$ otherwise.

This distinction gives rise to two classes of metrics: ranking-based metrics defined on $\hat{p}_t$, and threshold-dependent metrics defined on $\hat{y}_t$.

6.2. Discrimination metrics

The benchmark reports both ROC-AUC and PR-AUC. ROC-AUC measures a model's ability to rank positive and negative observations correctly across all possible thresholds, summarizing the tradeoff between the true positive rate and the false positive rate. It remains a widely used measure of general ranking discrimination.

PR-AUC summarizes the tradeoff between precision and recall across thresholds. In imbalanced classification problems, PR-AUC is often more informative than ROC-AUC because it focuses directly on the quality of positive predictions [17,19]. Since drawdown events are relatively infrequent in the present setting, PR-AUC plays a central role in the benchmark and serves as the main basis for many of the ranking comparisons reported later in the paper.

For a given threshold τ, the confusion matrix consists of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). Based on these quantities, precision and recall are defined as

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \qquad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}.$$

Precision measures the fraction of alerts that are correct, whereas recall measures the fraction of realized positive observations that are successfully detected. The F_1 score combines these through their harmonic mean:

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$

In the benchmark, precision, recall, and F_1 are used both as reported threshold-dependent metrics and, under one benchmark specification, as part of the threshold-selection rule.

6.3. Operational metrics

Because an early-warning system is ultimately judged by how it behaves as a warning mechanism, the benchmark also reports operational metrics. False alarms per year measures how often the model issues warnings that are not followed by a realized market-stress event. This metric is important because a warning system that signals too frequently may become impractical even if it performs reasonably well on conventional classification measures.

Event hit rate measures the fraction of realized stress episodes successfully captured by the warning system under the selected threshold rule. Unlike observation-level recall, event hit rate is defined at the episode level and is intended to reflect whether the model identifies the distinct stress events that matter from a monitoring perspective.

 doi:10.20944/preprints202603.1925.v1

Median lead summarizes how early, in successful detections, the warning signal arrives relative to the onset of a realized stress event. Lead time matters because the objective of early warning is not merely to coincide with stress, but to identify elevated downside risk before the event is fully realized. The benchmark reports median lead rather than mean lead in order to reduce sensitivity to a small number of unusually early or unusually delayed detections.

6.4. Alert-budget evaluation

In practical monitoring environments, users often face an implicit alert budget and may be able to review only a limited number of high-risk warnings. For that reason, the benchmark includes an alert-budget evaluation based on top-fraction ranking performance.

Let q denote the fraction of days assigned the highest predicted risk scores. For each value of q, the benchmark evaluates precision and recall within this top-risk subset. This provides an additional operational view of model usefulness by asking how informative the strongest warnings are when attention is limited to a relatively small number of alert days.

6.5. Cost-sensitive interpretation of operational tradeoffs

The benchmark's primary evaluation relies on discrimination and operational monitoring metrics rather than on a fully specified utility or portfolio objective. Even so, these operational metrics can be interpreted through a simple cost-sensitive lens. In practical monitoring environments, missed stress episodes and unnecessary alerts need not carry the same cost.

For model m, define

$$L_m(\lambda) = \lambda \cdot \text{MissedEvents}_m + \text{FApy}_m,$$

where MissedEvents_m denotes the number of realized stress episodes not captured by model m, FApy_m denotes false alarms per year, and $\lambda > 0$ represents the relative cost of missing one stress episode compared with issuing an unnecessary alert. Larger values of λ correspond to more crisis-averse monitoring preferences.

This calculation is not intended to estimate a universal economic loss function. Rather, it provides a transparent way to interpret operational tradeoffs under alternative monitoring priorities. In the empirical results, the loss is evaluated for several illustrative values of λ in order to show whether model preference changes when missed stress episodes are weighted more heavily than false alarms.

6.6. Metric roles in the benchmark

The benchmark does not treat all metrics as interchangeable. PR-AUC and ROC-AUC evaluate ranking discrimination independently of a specific binary threshold, with particular emphasis on PR-AUC because of the rare-event nature of the problem. Precision, recall, and F_1 summarize threshold-dependent classification quality. False alarms per year, event hit rate, and median lead translate predictive performance into operational terms relevant for real-time monitoring. Alert-budget analysis adds a further lens by examining the informativeness of the model's highest-risk predictions under limited attention.

Because the purpose of the paper is to study benchmark sensitivity, these metrics are used not only to assess performance levels but also to evaluate ranking stability across specifications. A model that ranks first under one specification may lose that advantage under a different feature set, horizon, walk-forward mode, or threshold rule. The benchmark therefore considers both performance within a specification and stability across specifications.

7. Results

This section reports the empirical findings from the leakage-safe benchmark. The objective is not to identify a universally dominant model under a single preferred specification, but to assess how model rankings change across benchmark designs and whether those rankings remain stable when both statistical and operational criteria are considered. Three main findings emerge. First, the

 doi:10.20944/preprints202603.1925.v1

benchmark is specification-sensitive: model rankings vary across feature sets, forecast horizons, walk-forward modes, drawdown thresholds, and threshold rules. Second, Random Forest performs best on average in terms of relative ranking and event-detection robustness across the main benchmark grid. Third, XGBoost is often more conservative operationally, generating fewer false alarms, but its relative standing is more dependent on specification choice, particularly the feature set and walk-forward design.

7.1. Primary benchmark specification

The primary benchmark specification is defined as `cross_asset`, forecast horizon $H = 63$, drawdown threshold 0.08, rolling walk-forward evaluation, and the `precision_target` threshold rule. Figure 1 summarizes the model comparison under this reference setup, and Table 2 reports the corresponding numerical values.

Under the primary specification, Random Forest delivers the strongest statistical performance. Its PR-AUC is 0.0161, compared with 0.0153 for XGBoost and 0.0114 for the volatility baseline. The same ordering appears in ROC-AUC, where Random Forest attains 0.5232, XGBoost 0.5113, and the volatility baseline 0.3602. Although the gap between the two machine-learning models is not large, the ordering is directionally consistent.

The operational comparison is more nuanced. Random Forest and the volatility baseline both achieve an event hit rate of 0.4444, whereas XGBoost records 0.3333. However, XGBoost generates materially fewer false alarms, with 11.6812 false alarms per year compared with 21.0831 for Random Forest and 23.0774 for the volatility baseline. Random Forest therefore combines stronger discrimination with better event coverage, but at the cost of a heavier alert burden. The volatility baseline does not offset its weaker statistical performance with a sufficient operational advantage.

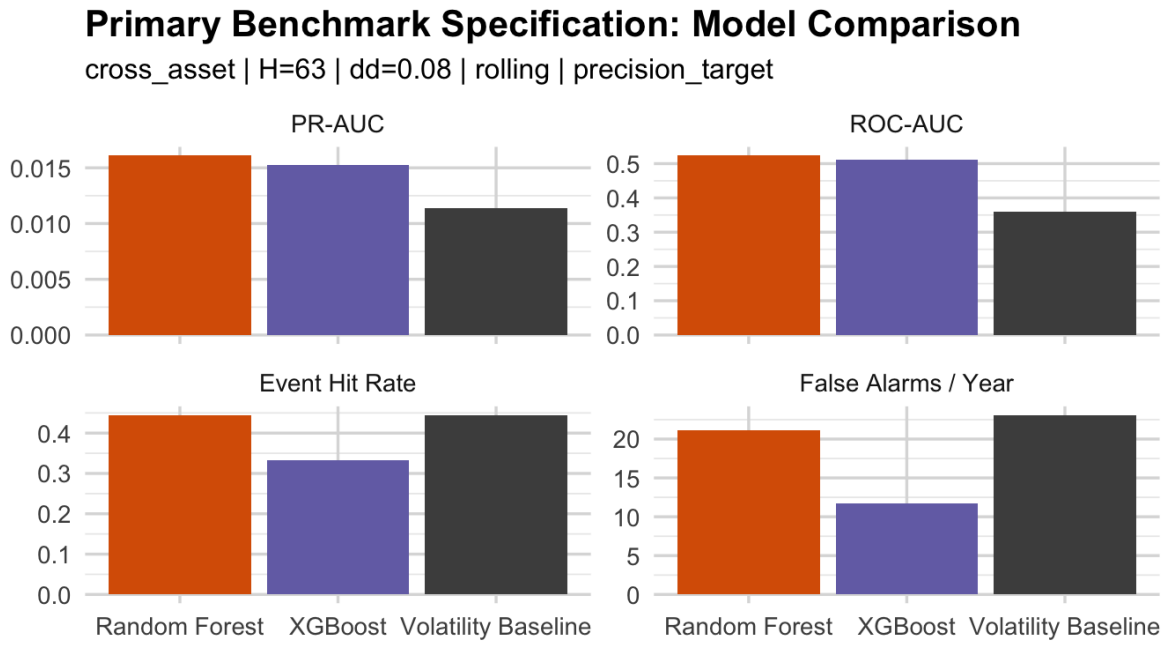

Figure 1. Primary benchmark specification model comparison.

Table 2. Primary specification results.

Model	PR-AUC	ROC-AUC	Event hit rate	False alarms/year	Median lead
Random Forest	0.0161	0.5232	0.4444	21.0831	33.5
XGBoost	0.0153	0.5113	0.3333	11.6812	NA
Volatility Baseline	0.0114	0.3602	0.4444	23.0774	NA

 doi:10.20944/preprints202603.1925.v1

7.2. Ranking sensitivity across benchmark specifications

The benchmark includes 32 main specifications formed from the combination of two feature sets, two forecast horizons, two drawdown thresholds, two walk-forward modes, and two threshold rules. Figure 2 visualizes the PR-AUC rank of each model across these specifications.

The main result is that model rankings are not invariant across the design grid. Random Forest is more frequently near the top of the ranking, whereas XGBoost alternates between strong and moderate positions depending on the specification. The volatility baseline is generally ranked below the machine-learning models, although it remains informative as a transparent market-based comparator.

This ranking sensitivity is central to the paper's contribution. A single-specification comparison could easily overstate the apparent superiority of one model. By contrast, the heatmap shows that conclusions are conditional on how the evaluation problem is defined. The empirical question is therefore not simply which model performs best, but how robust that conclusion is to plausible benchmark variation.

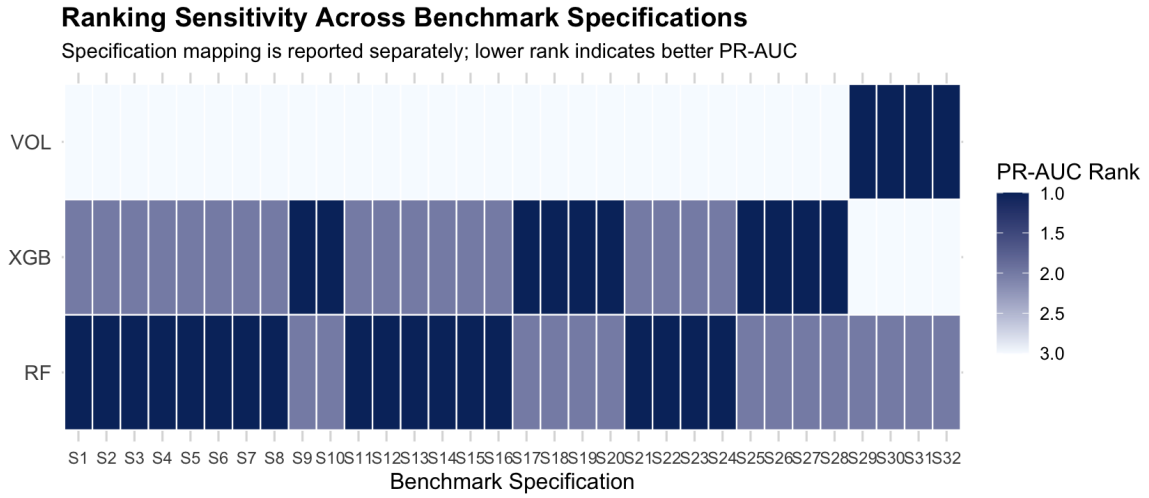

Figure 2. Ranking sensitivity across benchmark specifications. Lower rank indicates better PR-AUC within each specification.

7.3. Average ranking and benchmark stability

Table 3 and Figure 3 summarize average performance across the full benchmark grid. Random Forest has the best average PR-AUC rank, at 1.4375, and the best average event-hit-rate rank, at 1.3281. XGBoost ranks second on both dimensions, with an average PR-AUC rank of 1.8125 and an average hit-rate rank of 2.3438. The volatility baseline is weaker overall, with an average PR-AUC rank of 2.7500.

The main counterpoint is false-alarm burden. XGBoost achieves the best average false-alarm rank, at 1.2188, clearly outperforming Random Forest at 1.8438 and the volatility baseline at 2.9375. This confirms a recurring pattern in the benchmark: Random Forest is stronger in ranking quality and event detection, whereas XGBoost is more conservative in alert generation.

The count summaries reinforce this interpretation. Random Forest ranks first in PR-AUC in 18 of the 32 specifications and achieves the strongest event-hit-rate count in 19 specifications. XGBoost ranks first in PR-AUC in 10 specifications but has the lowest false-alarm count in 23 specifications. The volatility baseline ranks first only occasionally and remains weakest overall.

 doi:10.20944/preprints202603.1925.v1

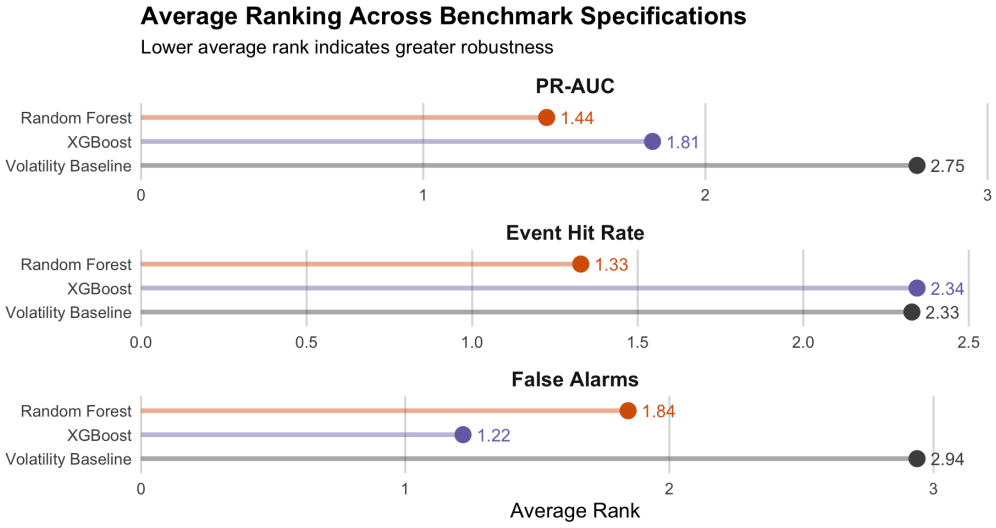

Figure 3. Average ranking across benchmark specifications. Lower average rank indicates greater robustness.

Table 3. Stability and fragility summary across the main benchmark specifications.

Model	PR range	ROC range	Hit range	FA range	SD(PR rank)	SD(Hit rank)	SD(FA rank)
Random Forest	0.0268	0.6354	0.7500	54.9869	0.5040	0.4327	0.4826
XGBoost	0.0519	0.5452	0.4444	19.3736	0.6445	0.4655	0.3797
Volatility Baseline	0.0080	0.0916	0.5000	70.2157	0.6720	0.5765	0.2459

Notes: "FA" denotes false alarms. Lower rank standard deviations indicate greater stability across benchmark specifications.

To distinguish robustness from fragility, Table 3 reports the range of outcomes and the dispersion of ranks across specifications. Random Forest exhibits the lowest standard deviation of PR-AUC rank, at 0.5040, and the lowest standard deviation of hit-rate rank, at 0.4327. This indicates that its relative standing is more stable than that of the other two models. XGBoost has a wider PR-AUC range, at 0.0519, and a higher PR-AUC rank standard deviation, at 0.6445, indicating greater sensitivity to benchmark design. The volatility baseline shows a narrow PR-AUC range but remains weak across most specifications, which suggests stable underperformance rather than competitive robustness.

7.4. Robustness and operational trade-offs

Figure 4 provides a direct view of PR-AUC sensitivity across forecast horizon, drawdown threshold, feature set, walk mode, and threshold rule. Performance is visibly stronger when the drawdown threshold is set at 0.08 rather than 0.10. Under the stricter threshold, many specifications collapse toward very low PR-AUC values, reflecting greater event sparsity and a more difficult discrimination problem.

Across the grid, Random Forest is more consistently competitive, whereas XGBoost shows greater upside in some settings, particularly under richer information sets. The volatility baseline remains weaker throughout, even when its values vary less across specifications. Low dispersion should therefore not be interpreted as robustness when the overall performance level is weak.

Figure 5 highlights the core operational trade-off by plotting false alarms per year against event hit rate, with point size reflecting PR-AUC. Random Forest tends to occupy the upper-middle part of the plot, indicating stronger event detection but also a heavier alert burden. XGBoost lies further to the left in many specifications, reflecting fewer false alarms but often lower event hit rates. The volatility baseline is frequently dominated, combining relatively weak PR-AUC with either high false alarms or only modest event coverage.

Taken together, these figures show why discrimination metrics alone are insufficient. A model can appear attractive in terms of PR-AUC yet impose a large alert burden, while a conservative model

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 March 2026 doi:10.20944/preprints202603.1925.v1

can reduce false alarms at the cost of weaker event detection. The benchmark therefore supports a multidimensional interpretation rather than a single-metric comparison.

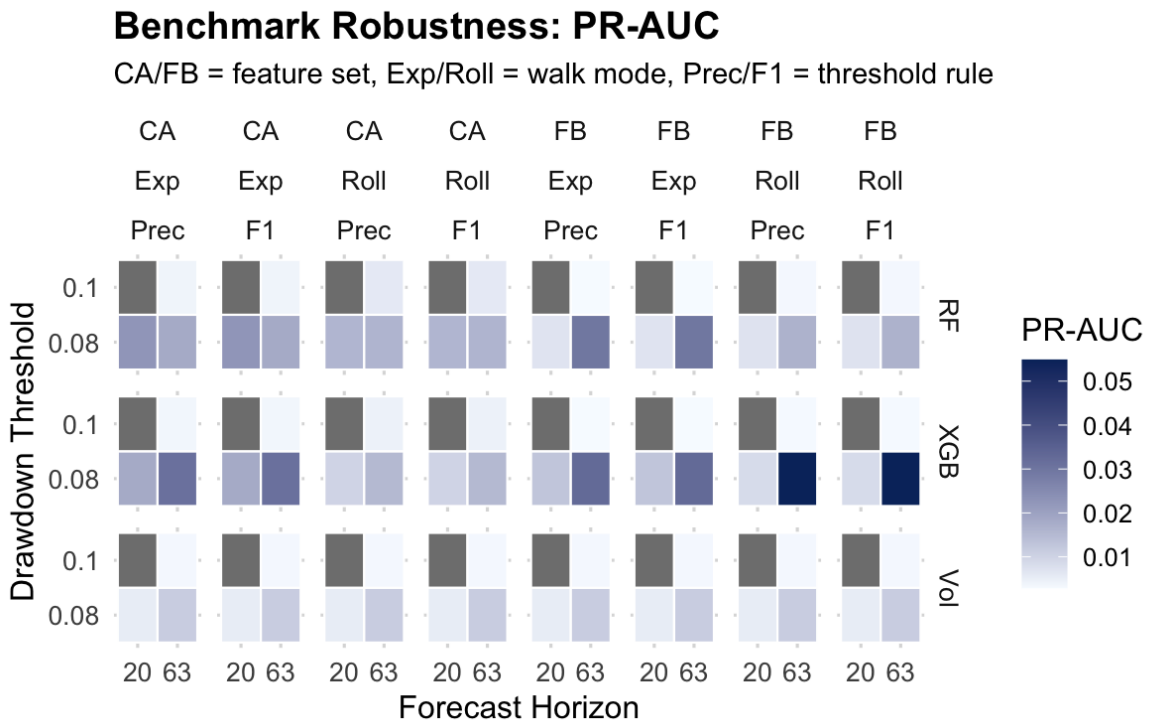

Figure 4. PR-AUC robustness across forecast horizon, drawdown threshold, feature set, walk mode, and threshold rule.

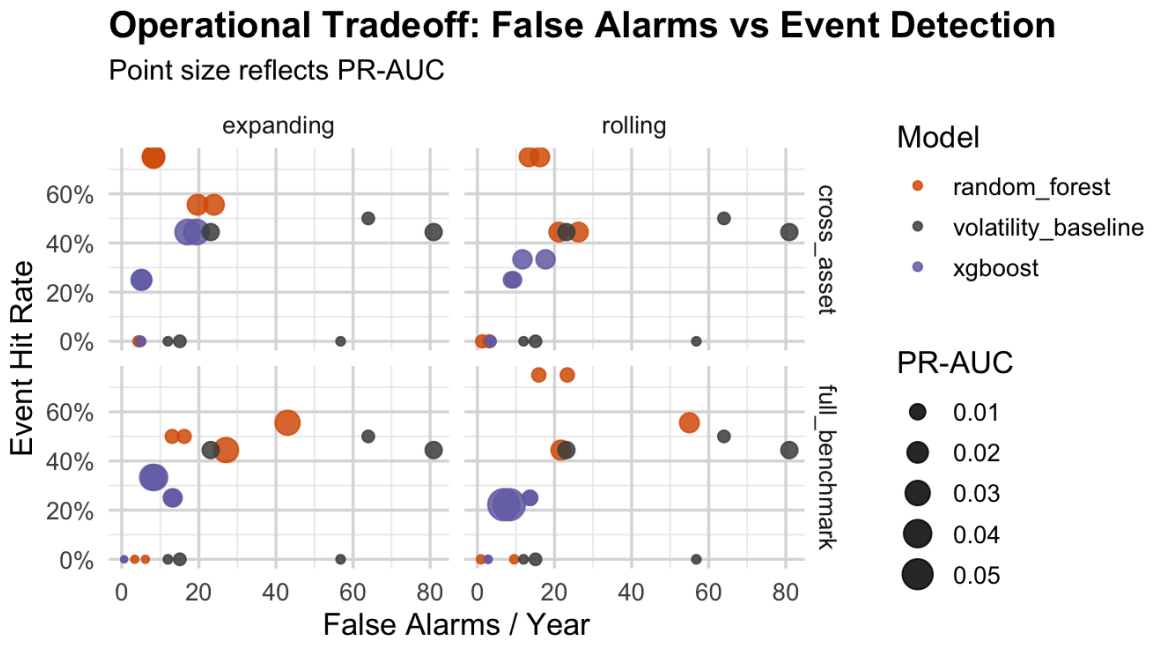

Figure 5. Operational tradeoff between false alarms per year and event hit rate. Point size reflects PR-AUC.

7.5. Cost-sensitive interpretation

Table 4 reports a simple cost-sensitive comparison under the primary benchmark specification, which contains 9 realized stress episodes. Under this setup, Random Forest and the volatility baseline each miss 5 episodes, whereas XGBoost misses 6. When false alarms and missed events receive only modestly different weights, XGBoost remains attractive because of its lower alert burden. As the

 doi:10.20944/preprints202603.1925.v1

relative cost of missed stress episodes increases, however, the stronger event-detection performance of Random Forest becomes more valuable and the gap between the two models narrows materially. Under sufficiently crisis-averse preferences, Random Forest becomes competitive despite producing more false alarms. The volatility baseline remains least attractive overall because it combines weaker discrimination with the highest alert burden.

Table 4. Cost-sensitive comparison under the primary benchmark specification.

Model	False alarms/year	Missed events	Loss ($\lambda = 2$)	Loss ($\lambda = 5$)	Loss ($\lambda = 10$)
Random Forest	21.0831	5	31.0831	46.0831	71.0831
XGBoost	11.6812	6	23.6812	41.6812	71.6812
Volatility Baseline	23.0774	5	33.0774	48.0774	73.0774

7.6. Sensitivity to benchmark dimensions

Figures 6–8 isolate the effects of individual benchmark dimensions. Figure 6 shows that sensitivity to the threshold-selection rule is modest in PR-AUC terms, which is expected because PR-AUC is ranking-based and threshold selection primarily affects the conversion from probabilities to binary alerts. Even so, threshold choice remains operationally important because it influences precision, recall, event hit rate, and false alarms.

Figure 7 shows that walk-mode choice materially affects the machine-learning models, especially XGBoost. Under the expanding design, XGBoost attains noticeably higher PR-AUC than under the rolling design. Random Forest also improves under expanding evaluation, although less sharply. By contrast, the volatility baseline changes little across walk modes. This suggests that the more flexible models are more sensitive to how historical information is accumulated and weighted over time.

Figure 8 shows one of the clearest examples of benchmark dependence. Random Forest changes only modestly across the `cross_asset` and `full_benchmark` feature sets, and the volatility baseline remains similarly weak in both. XGBoost, however, improves sharply under the richer `full_benchmark` specification. This suggests that its relative competitiveness depends more strongly on predictor richness than does that of Random Forest.

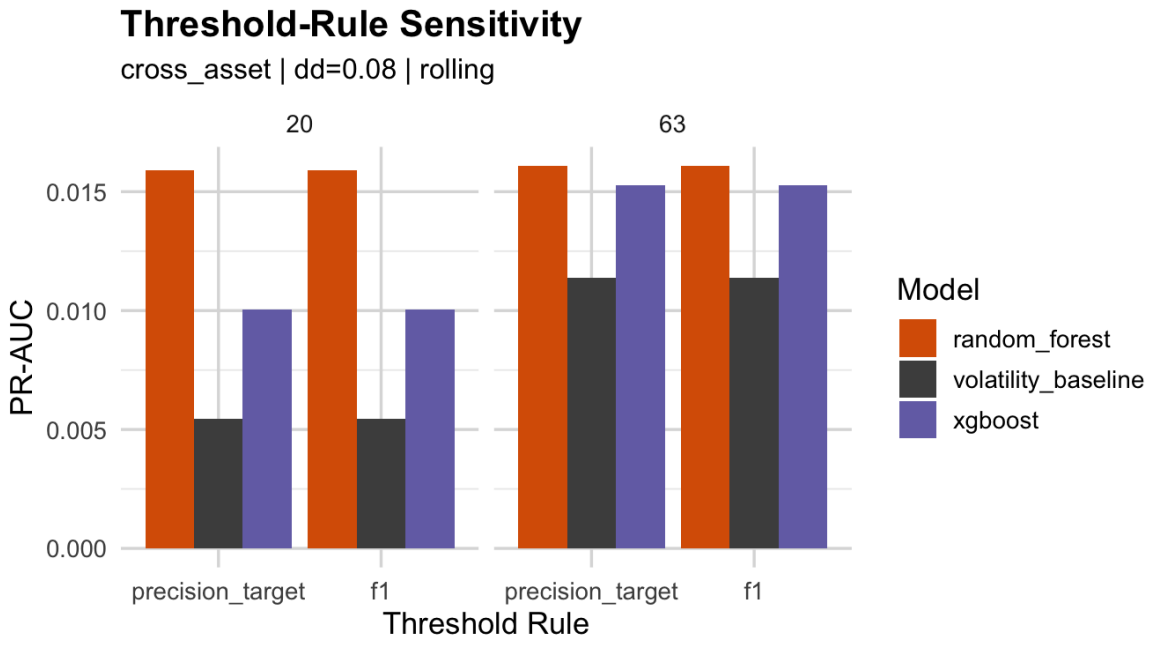

Figure 6. Threshold-rule sensitivity for the `cross_asset` feature set at drawdown threshold 0.08 under rolling walk-forward evaluation.

 doi:10.20944/preprints202603.1925.v1

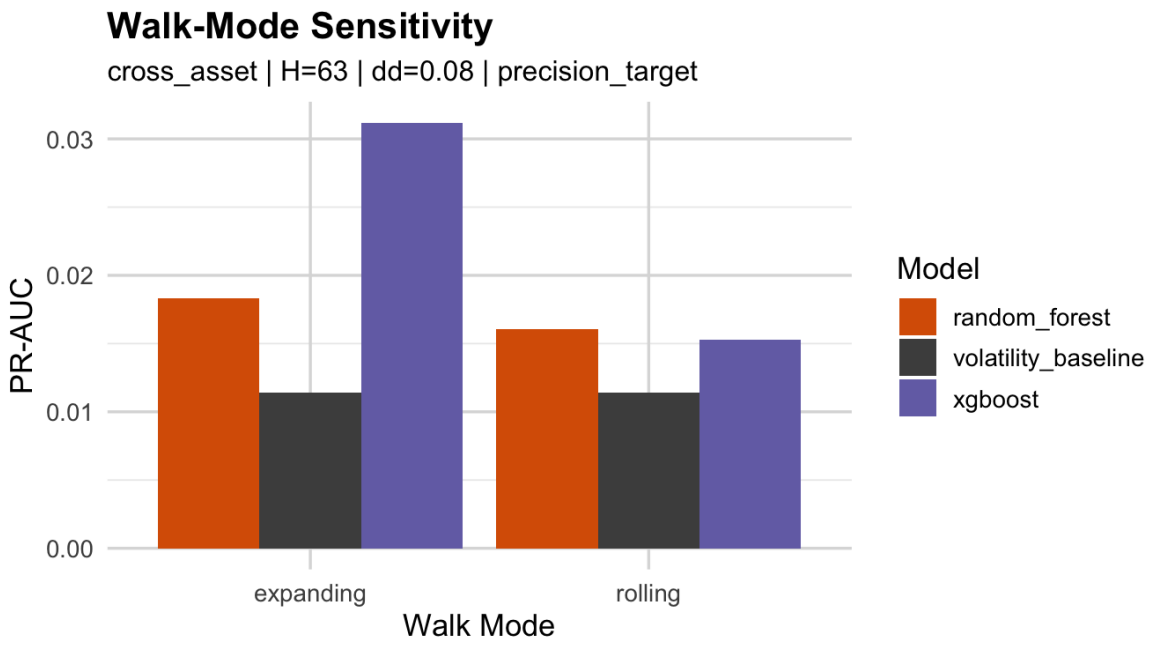

Figure 7. Walk-mode sensitivity under the primary horizon, drawdown threshold, and threshold rule.

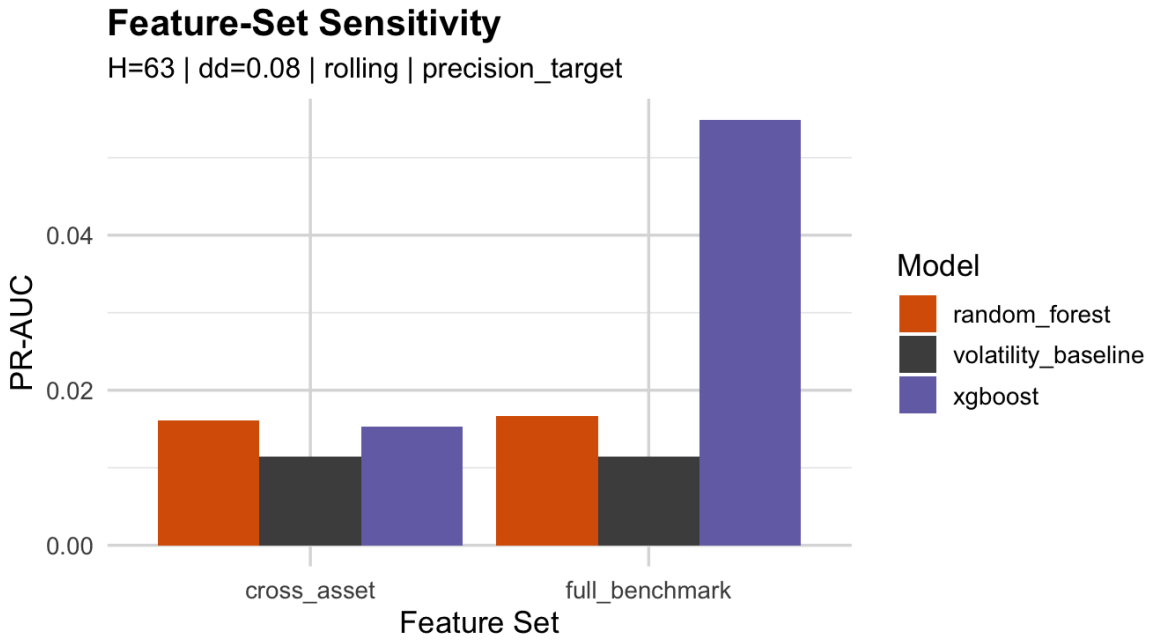

Figure 8. Feature-set sensitivity under the primary horizon, drawdown threshold, walk mode, and threshold rule.

7.7. Stylized defensive-monitoring interpretation

The benchmark results above assess model quality using both discrimination-based metrics and operational monitoring criteria. To connect these results to a more intuitive economic interpretation, this subsection introduces a stylized defensive-monitoring lens. The objective is not to define or optimize a trading strategy, but to clarify how differences in alerts, missed events, and false alarms map into capital-preservation trade-offs.

Suppose that when model m issues an alert at date t, a fraction $\omega \in [0,1]$ of the risky position is temporarily shifted into cash for the forecast horizon H, while the remaining fraction $1-\omega$ stays invested in the risky asset. Let $R_{t,t+H}$ denote the cumulative return on the risky asset over the next H

 doi:10.20944/preprints202603.1925.v1

trading days, and let $R^c_{t,t+H}$ denote the corresponding cash return. For short horizons, $R^c_{t,t+H}$ is set to zero for simplicity. The resulting stylized defensive return is

$$R^{\text{def}}_{t,t+H} = (1-\omega)R_{t,t+H} + \omega R^c_{t,t+H} \approx (1-\omega)R_{t,t+H}. \tag{1}$$

This construction is intentionally illustrative rather than implementational. Its purpose is to show how an alert may be interpreted as a temporary partial de-risking decision. If alerts are informative, then dates flagged by the model should be followed, on average, by weaker forward returns and worse forward drawdowns than non-alert dates. Let $\mathcal{A}_m$ denote the set of alert dates and $\mathcal{N}_m$ the set of non-alert dates. The corresponding average forward outcomes are

$$\bar{R}^A_m = \frac{1}{|\mathcal{A}_m|} \sum_{t \in \mathcal{A}_m} R_{t,t+H}, \tag{2}$$

$$\bar{R}^N_m = \frac{1}{|\mathcal{N}_m|} \sum_{t \in \mathcal{N}_m} R_{t,t+H}. \tag{3}$$

The average gain from taking the stylized defensive action on alert dates is

$$\Delta^{\text{def}}_m = \frac{1}{|\mathcal{A}_m|} \sum_{t \in \mathcal{A}_m} \left(R^{\text{def}}_{t,t+H} - R_{t,t+H} \right). \tag{4}$$

When forward returns following alerts are negative on average, Δ^{def}_m is positive, indicating that partial de-risking would improve outcomes under this stylized rule. When forward returns following alerts are positive, Δ^{def}_m becomes negative, reflecting the opportunity cost of an unnecessary defensive shift.

A complementary event-oriented interpretation follows from the cost-sensitive loss

$$L_m(\lambda) = \lambda \cdot \text{MissedEvents}_m + \text{FApy}_m, \tag{5}$$

where MissedEvents_m is the number of stress episodes not covered by model m, FApy_m denotes false alarms per year, and $\lambda > 0$ captures the relative cost of missing a stress episode. Larger values of λ correspond to more crisis-averse users who are willing to tolerate more false alarms in exchange for stronger event coverage.

Under this interpretation, three implications follow. First, warning systems with stronger event coverage become more attractive when missed stress episodes are especially costly. Second, more conservative systems reduce unnecessary defensive actions but may also leave more stress episodes uncovered. Third, the economic appeal of a warning system remains benchmark-dependent because users may place different weights on missed events, false alarms, and monitoring stability.

7.8. Main empirical takeaway

Taken together, the results support three broad conclusions. First, Random Forest exhibits the strongest overall relative pattern within this benchmark when performance is judged by average ranking and event-detection robustness across the benchmark grid. Second, XGBoost is often more conservative operationally, with substantially fewer false alarms, but it is also more sensitive to benchmark design, especially feature-set richness and walk-mode choice. Third, the volatility baseline remains a useful comparator but is generally weaker than the machine-learning models in this benchmark and does not deliver a sufficient operational advantage to offset its weaker discrimination on average.

More broadly, the results support the paper's central claim: conclusions about model quality in market-stress early warning depend materially on benchmark design. Feature set, forecast horizon, drawdown threshold, walk-forward scheme, and threshold-selection rule all affect the empirical ordering. The main contribution of the benchmark is therefore not to declare a universal winner, but to show when apparent model superiority is relatively robust and when it remains conditional on how the evaluation problem is constructed.

 doi:10.20944/preprints202603.1925.v1

8. Reproducible Research Pipeline

Section 3.7 summarizes the paper's reproducibility principles. This section explains how those principles are implemented in the empirical workflow used to generate the reported benchmark results.

The analysis pipeline is organized so that each reported result is linked to a benchmark specification indexed by feature set, forecast horizon, drawdown threshold, walk-forward mode, and threshold-selection rule. This structure allows the full specification grid to be evaluated consistently and makes it possible to study robustness within a common empirical design rather than through a sequence of ad hoc reruns.

At a practical level, the manuscript analysis begins from saved outputs generated by the leakage-safe evaluation procedure. These outputs include pooled summary tables, alert-budget summaries, robustness summaries, and row-level test predictions where available. The workflow then validates the required identifiers and metric fields and constructs the derived objects used in the paper, including the primary-specification comparison table, ranking summaries, stability diagnostics, and figure-ready panels.

This design is important because the paper's tables and figures are generated directly from benchmark outputs rather than assembled manually. As a result, the primary comparison figure, ranking heatmap, robustness heatmap, operational trade-off plot, average-ranking summary, and related sensitivity figures are all downstream products of the same empirical workflow. If an upstream benchmark output changes, the corresponding manuscript results can be regenerated consistently.

The same structure also makes it possible to distinguish between main-text and appendix analyses without introducing separate analytical logic. The main text focuses on the three primary models—Random Forest, XGBoost, and the volatility baseline—whereas the appendix extends the comparison to auxiliary baselines. Because both layers are derived from the same workflow, the distinction between core and supplementary evidence is handled through explicit filtering and aggregation rather than through separate implementations.

More broadly, this implementation structure supports the economic credibility of the benchmark comparison. In leakage-sensitive financial applications, seemingly minor differences in filtering, metric mapping, or plotting logic can alter reported out-of-sample conclusions. Centralizing benchmark logic in code therefore helps reduce silent inconsistencies and improves the transparency and auditability of the empirical evidence.

Thus, the role of the pipeline is not to redefine the benchmark design introduced earlier, but to ensure that the reported comparison remains systematically linked to the underlying leakage-safe evaluation results. In this sense, reproducible implementation is part of credible computational evaluation, because it helps preserve the interpretability and comparability of the evidence across benchmark specifications.

In summary, the research pipeline supports the analysis in three ways: it links benchmark specifications to saved evaluation outputs in a structured manner, it generates manuscript tables and figures directly from those outputs, and it facilitates transparent extension to additional models, feature sets, and operational metrics.

9. Discussion and Limitations

The results support a benchmark-centered interpretation of market-stress early warning. The main contribution is not the identification of a universally dominant forecasting model, but evidence that conclusions about model quality depend materially on benchmark design. Across the specification grid, rankings change with the feature set, forecast horizon, drawdown threshold, walk-forward scheme, and threshold-selection rule. This point matters because results that appear strong under one benchmark configuration may not generalize even to nearby and still plausible alternatives. Isolated out-of-sample victories should therefore be interpreted with caution.

A second implication is that statistical discrimination and operational usefulness are not interchangeable. In the benchmark, Random Forest performs best on average in terms of relative ranking

 doi:10.20944/preprints202603.1925.v1

and event-detection robustness, whereas XGBoost more often delivers a lower false-alarm burden. These findings are not contradictory. They reflect the fact that early-warning systems are evaluated under multiple objectives. A model with stronger PR-AUC or ROC-AUC may still be less attractive in deployment if it generates too many alerts, while a more conservative model may fail to capture enough stress episodes to be useful. Model evaluation in this setting should therefore be aligned explicitly with the intended monitoring use case rather than reduced to a single summary metric.

More broadly, the benchmark grid can be interpreted as a structured way to translate statistical performance into alternative economic monitoring preferences. Users with a low tolerance for missed stress episodes may prefer a system with a heavier alert burden, whereas users facing higher operational costs from false alarms may prefer a more conservative classifier. In this sense, threshold selection and benchmark specification are not merely technical tuning choices, but components of the broader economic design of a warning system.

The feature-set and walk-mode results point to a broader methodological lesson. XGBoost appears to benefit more strongly from a richer predictor universe and from certain walk-forward designs than Random Forest, indicating that model flexibility interacts with benchmark design in nontrivial ways. The answer to the question of which model performs best therefore depends partly on what information set is made available and how the time-series estimation problem is structured. In financial applications, where data richness, regime instability, and structural change all matter, such interactions are economically relevant rather than merely technical.

These findings should not be read as an argument against model comparison. On the contrary, the results show that meaningful comparison remains possible when the protocol is transparent and leakage-safe. The benchmark identifies clear regularities: Random Forest is strongest overall across the main grid, XGBoost is frequently the most conservative in terms of false alarms, and the volatility baseline is generally weaker than the machine-learning alternatives. The key point is that these patterns are better interpreted as robust tendencies than as universal laws. That distinction matters for both academic interpretation and practical deployment.

Several limitations define the scope of the study. First, the benchmark uses a deliberately selective model set rather than an exhaustive catalog of forecasting methods. Random Forest, XGBoost, and a volatility baseline provide a useful range of flexibility and interpretability, and the appendix broadens the comparison with several auxiliary baselines. Even so, the study does not evaluate every relevant machine-learning, econometric, or hybrid approach to market-stress prediction. Future work could extend the benchmark to additional model classes, especially methods tailored to rare-event detection or sequential decision settings.

Second, the results are conditional on the event definition adopted here. Stress events are defined through forward drawdowns over fixed horizons and thresholds, which is appropriate for the present objective but not unique. Alternative definitions based on volatility bursts, liquidity deterioration, tail-return exceedances, or regime-switching conditions could produce different comparative results. The evidence should therefore be understood as applying to an economically meaningful but still particular family of early-warning targets.

Third, some benchmark configurations operate in sparse-event environments, especially when the drawdown threshold is set at 0.10. In those cases, several threshold-dependent metrics become zero, undefined, or weakly identified. This does not invalidate the benchmark, but it does limit the precision with which performance can be compared under the most severe event definitions. The presence of these cases is itself informative, since it shows how quickly benchmark difficulty rises as the target becomes rarer.

Fourth, although the benchmark emphasizes reproducibility and leakage safety, no empirical protocol can remove all researcher discretion. Feature engineering, sample choice, tuning decisions, and reporting conventions still matter. The present study addresses that issue by making benchmark design explicit and by treating robustness analysis as part of the main contribution. Even so, the reported evidence remains conditional on the selected empirical design.

 doi:10.20944/preprints202603.1925.v1

Fifth, the operational metrics improve the practical interpretation of the benchmark, and the cost-sensitive comparison helps connect monitoring performance to economically meaningful tradeoffs. By showing how model preference changes when missed stress episodes receive greater weight than false alarms, the analysis moves beyond purely descriptive operational summaries and offers a decision-oriented interpretation of monitoring performance. Even so, the benchmark does not yet embed the warning system in a full decision-theoretic or portfolio framework with realized allocation outcomes. False alarms, event hit rates, and lead times remain informative operational summaries, but they do not fully encode the economic costs of defensive actions, capital-allocation responses, or institutional constraints. A natural extension would therefore be to build richer utility-based, capital-preservation, or portfolio-linked evaluation rules on top of the benchmark developed here.

Sixth, the stylized defensive-monitoring interpretation introduced in the results is intended as a conceptual bridge between warning-system quality and economic relevance rather than as a fully implemented policy simulation. It clarifies how alert behavior may be interpreted through a capital-preservation lens, but it is not designed to establish implementable portfolio superiority, optimal execution, or realized allocation performance. Future work could extend this perspective by linking the warning signals to explicit defensive rules, transaction-cost assumptions, dynamic rebalancing, or institutional response constraints.

Finally, the reproducible pipeline strengthens transparency, but practical reproducibility also depends on data access, software environments, and version control. Even a released codebase requires documentation and dependency management before other researchers can reproduce the full set of results without ambiguity. The present study should therefore be viewed as an important support for transparent benchmark construction rather than as a complete solution to every implementation challenge associated with replication.

These limitations do not weaken the central contribution of the paper; they define its scope. The study does not claim to provide a universal market-stress detector, a complete ranking of all forecasting methods, or a final event definition for early-warning research. Its contribution is instead to show that leakage-safe evaluation, operationally meaningful performance criteria, cost-sensitive interpretation, and transparent benchmark design are all necessary for economically credible comparison in market-stress early warning. The main implication is that benchmark design is itself a substantive determinant of model usefulness, because it shapes the conclusions drawn from computational comparison in financially relevant monitoring settings.

10. Conclusions

This paper studies market-stress early warning as a problem of economically credible model comparison in temporally ordered financial settings. Using a leakage-safe out-of-sample benchmark design, the analysis examines whether conclusions about model usefulness remain stable across alternative feature sets, forecast horizons, drawdown thresholds, walk-forward schemes, and threshold-selection rules. Performance is evaluated not only with standard discrimination measures, but also with operational criteria that are directly relevant to monitoring decisions, including false alarms, event coverage, lead time, and alert-budget tradeoffs.

The main empirical result is that benchmark design materially affects both model rankings and the economic interpretation of predictive usefulness. Across specifications, the relative ordering of Random Forest, XGBoost, and the volatility baseline changes, and models that perform well on conventional statistical metrics are not always preferred once alert burden, event coverage, and usable lead time are taken into account. Within the benchmark considered here, Random Forest performs most strongly on average and appears most robust in event detection across the main specifications. XGBoost is often more conservative, generating fewer false alarms, but its relative standing is less stable across benchmark designs. The volatility baseline remains an important and transparent reference point, but it is generally weaker than the machine-learning alternatives in overall operational performance.

 doi:10.20944/preprints202603.1925.v1

The broader implication is that benchmark specification should be treated as part of the empirical economics problem rather than as a neutral implementation detail. Choices regarding feature construction, forecast horizon, event definition, walk-forward design, and decision threshold shape the conclusions drawn about which warning systems are economically useful. A system that appears attractive under one specification may become materially less attractive under another once the trade-off between missed events and unnecessary alerts is made explicit. The cost-sensitive comparisons reinforce this point by showing that model preference depends on the monitoring objective and on the relative weight assigned to missed stress episodes versus false alarms.

This perspective is especially important in financial monitoring environments, where computational design choices can materially influence the evidence used to support real-time surveillance and risk assessment. By imposing a leakage-safe and transparent evaluation structure, the paper aims to make these comparisons more economically interpretable and more reliable. In this sense, the contribution is not only to compare alternative warning systems, but also to clarify how benchmark design governs the substantive conclusions of computational model evaluation in financial early-warning applications.

The analysis remains subject to several limitations. The benchmark is conditional on the candidate models, feature definitions, and drawdown-based event construction adopted here, and it does not exhaust the full range of early-warning designs or institutional decision environments. In the most severe event settings, outcome sparsity limits the stability of some threshold-dependent measures. In addition, although the cost-sensitive analysis improves decision relevance, the framework does not yet embed a fully specified decision-theoretic or portfolio-allocation objective. Accordingly, the contribution of the paper is not to claim a universally dominant warning system, but to provide a leakage-safe basis for economically credible comparison.

Overall, the evidence suggests that reliable market-stress early warning depends not only on model choice, but also on benchmark design. Economically meaningful comparison requires leakage-safe evaluation, operationally relevant performance criteria, and explicit treatment of the tradeoffs that govern monitoring decisions. On that basis, the paper contributes to the computational economics of financial early warning by showing that benchmark design is itself a substantive determinant of model usefulness.

Author Contributions: The author conceived the study, designed the methodology, implemented the experiments, analyzed the results, and wrote the manuscript.

Funding: The author received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw market data analyzed in this study are publicly available from Yahoo Finance. The processed datasets can be regenerated from the raw data using the analysis code described below. The code used to generate the results and figures is available from the author upon reasonable request.

Conflicts of Interest: The author declares no competing interests.

 doi:10.20944/preprints202603.1925.v1

References

1. Liu, T. Volatility Forecasting and Early-Warning Market Stress Detection: A Leakage-Safe Evaluation with Tree Ensembles and Transformers. *Preprint* **2026**, 04 March; available at Research Square. doi:10.21203/rs.3.rs-9015347/v1.
2. Merton, R. C. On the pricing of corporate debt: The risk structure of interest rates. *The Journal of Finance* **1974**, *29*(2), 449–470. doi:10.1111/j.1540-6261.1974.tb03058.x.
3. Campbell, J. Y.; Hilscher, J.; Szilagyi, J. In search of distress risk. *The Journal of Finance* **2008**, *63*(6), 2899–2939. doi:10.1111/j.1540-6261.2008.01416.x.
4. Aldasoro, I.; Borio, C.; Drehmann, M. Early warning indicators of banking crises: Expanding the family. *BIS Quarterly Review* **2018**, March, 29–45.
5. Liu, T. A comparative study of transformer-based and classical models for financial time-series forecasting. *Journal of Risk and Financial Management* **2026**, *19*(3), 203. doi:10.3390/jrfm19030203.
6. Chen, S.; Svirydzenka, K. Financial cycles—Early warning indicators of banking crises? *IMF Working Papers* **2021**, *2021*(116). doi:10.5089/9781513573895.001.
7. Aikman, D.; Haldane, A. G.; Nelson, B. D. Curbing the credit cycle. *The Economic Journal* **2015**, *125*(585), 1072–1109. doi:10.1111/ecoj.12113.
8. Andersen, T. G.; Bollerslev, T.; Diebold, F. X.; Labys, P. Modeling and forecasting realized volatility. *Econometrica* **2003**, *71*(2), 579–625.
9. Andersen, T. G.; Bollerslev, T.; Christoffersen, P. F.; Diebold, F. X. Practical volatility and correlation modeling for financial market risk management. *NBER Working Paper* **2005**, No. 11069. doi:10.3386/w11069.
10. Lo, A. W.; MacKinlay, A. C. Data-snooping biases in tests of financial asset pricing models. *The Review of Financial Studies* **1990**, *3*(3), 431–467. doi:10.1093/rfs/3.3.431.
11. Bailey, D. H.; Borwein, J. M.; López de Prado, M.; Zhu, Q. J. The probability of backtest overfitting. *Journal of Computational Finance* **2016**, *20*(4), 39–69. doi:10.21314/JCF.2016.311.
12. Bergmeir, C.; Hyndman, R. J.; Koo, B. A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics & Data Analysis* **2018**, *120*, 70–83.
13. Liu, T. Beyond volatility: A leakage-safe residual-stress signal for drawdown risk monitoring. *Preprints* **2026**. doi:10.20944/preprints202603.0395.v2.
14. Harutyunyan, H.; Khachatrian, H.; Kale, D. C.; Ver Steeg, G.; Galstyan, A. Multitask learning and benchmarking with clinical time series data. *Scientific Data* **2019**, *6*, 96. doi:10.1038/s41597-019-0103-9.
15. Li, H.; Liu, T. Portfolio optimization based on the LSTM forecasting model. *Proceedings of the 2nd International Conference on Financial Technology and Business Analysis* **2023**, *48*(1), 97–106.
16. Gu, S.; Kelly, B.; Xiu, D. Empirical asset pricing via machine learning. *The Review of Financial Studies* **2020**, *33*(5), 2223–2273.
17. Davis, J.; Goadrich, M. The relationship between precision-recall and ROC curves. In *Proceedings of the 23rd International Conference on Machine Learning*; ACM: New York, NY, USA, 2006; pp. 233–240.
18. Liu, T. Financial constraint impact on firms' ESG rating based on Chinese stock market. In *Proceedings of the 2022 4th International Conference on Economic Management and Cultural Industry (ICEMCI 2022)*; Atlantis Press: Paris, France, 2022; pp. 1085–1095.
19. Saito, T.; Rehmsmeier, M. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE* **2015**, *10*(3), e0118432. doi:10.1371/journal.pone.0118432.