Proceedings of the International Conference on Industrial Engineering and Operations Management

Publisher: IEOM Society International, USA DOI: 10.46254/NA11.20260215
Published: June 9, 2026

A Systematic Benchmark of Machine Learning Models for Last-Mile Delivery Delay Prediction with Interpretability Analysis

Arvin Bahreini
Lundquist College of Business
University of Oregon
Eugene, Oregon, USA
arvinb@uoregon.edu

Arman Nikkhah
Department of Computer Science
The University of Texas at Dallas
Richardson, Texas, USA
arman.nikkhah@utdallas.edu

Abstract

Last-mile delivery accounts for up to 53% of total supply chain costs, yet on-time delivery performance remains volatile due to weather disruptions, traffic congestion, and warehouse overloads. Proactive delay prediction offers logistics operators a principled path to early intervention. However, existing machine learning (ML) studies suffer from three persistent gaps: inconsistent benchmarking methodology, neglect of class imbalance (delays are inherently rare events), and opaque models that provide no actionable operational insights. This paper presents a systematic benchmark of three ML models—Logistic Regression (LR), Random Forest (RF), and XGBoost—evaluated on a synthetic 50,000-shipment last-mile delivery dataset with 12 operational features. Hyperparameters are tuned via randomized search with 5-fold cross-validation (CV), and robustness is assessed across five independent random seeds. We apply the Synthetic Minority Oversampling Technique (SMOTE), class-weight strategies, and scale_pos_weight to address imbalance, and use SHapley Additive exPlanations (SHAP) to interpret the best-performing model. After tuning, XGBoost achieves the highest test-set F1-score (0.516), Precision-Recall Area Under the Curve (PR-AUC) of 0.503, and Area Under the Receiver Operating Characteristic Curve (AUC-ROC) of 0.681, closely followed by LR (F1 = 0.514, AUC-ROC = 0.680). Multi-seed experiments confirm stable performance (XGBoost F1 = 0.513 ± 0.008; LR F1 = 0.513 ± 0.004). SHAP analysis identifies warehouse load factor, weather condition, and traffic index as the top three delay drivers, with rankings perfectly stable across three independent seeds. These findings deliver a reproducible, interpretable framework that logistics practitioners can deploy using routinely available Transportation Management System (TMS) data for proactive delay mitigation.

Keywords

Last-mile delivery, machine learning, XGBoost, SHAP interpretability, class imbalance

1. Introduction

The explosive growth of e-commerce has placed unprecedented strain on last-mile logistics—the final step of delivering a package from a regional distribution center to the end customer. According to industry estimates, last-mile delivery accounts for 53% of total supply chain costs and up to 40% of parcel carbon emissions (Gevaers et al.,

2011). In the United States alone, annual last-mile delivery volumes exceeded 165 billion packages in 2023 (Wu et al., 2023). Despite this scale, on-time delivery rates remain volatile: weather disruptions, traffic congestion, warehouse overloads, and demand surges regularly push delay rates above 20%, triggering financial penalties, customer churn, and reputational damage (Thomas and Panicker, 2023).

Proactive delay prediction—identifying shipments at risk before they miss their delivery window—offers a principled path to mitigation. If a logistics operator can predict with sufficient accuracy which deliveries will be delayed, resources can be reallocated, customer communications can be triggered, and alternative routing can be activated before the problem becomes visible to the customer. ML provides the predictive power to operationalize this capability from routinely collected operational data.

However, the existing literature on ML-based delivery delay prediction suffers from three compounding gaps. First, comparative studies typically evaluate one or two models under varying experimental conditions, making fair assessment of relative performance impossible (Baryannis et al., 2019). Second, delays are inherently rare events (typically 15–35% of shipments), creating class imbalance that causes naive classifiers to achieve high accuracy by simply predicting "on-time" for every shipment while completely failing to identify actual delays (Lesmana et al., 2025). Third, the highest-performing models (RF, XGBoost) are opaque to practitioners: logistics managers cannot act on a probability score if they do not understand what features drive it (Baryannis et al., 2019).

1.1 Motivation and Research Objectives

The operational cost of missed delivery predictions is significant: each undetected delay triggers downstream consequences including penalty fees, customer escalations, and reactive re-routing that is consistently more expensive than proactive intervention. Existing studies have not provided logistics practitioners with a reliable, reproducible framework that simultaneously addresses benchmarking rigor, class imbalance, and model transparency. This gap is particularly acute for operations managers who must justify model deployment decisions based on interpretable evidence rather than opaque performance scores.

This paper addresses all three gaps through the following contributions:

1. Systematic benchmark: We evaluate LR, RF, and XGBoost under identical experimental conditions with hyperparameter tuning via randomized search (5-fold CV), multi-seed robustness analysis (five seeds), and comprehensive metrics including F1, AUC-ROC, PR-AUC, and Brier score.
2. Class imbalance analysis: We compare ten configurations spanning SMOTE, class weighting, scale_pos_weight, and their combinations across all three model families, quantifying the recall improvement from imbalance handling.
3. SHAP interpretability: We apply TreeSHAP (Lundberg and Lee, 2017) to decompose XGBoost predictions into feature-level attributions, identifying actionable operational insights, and validate SHAP ranking stability across three independent seeds.
4. Reproducibility: We release all code and a synthetic 50,000-shipment dataset with fully documented generation parameters, enabling immediate replication and adaptation by logistics practitioners.

The remainder of this paper is organized as follows: Section 2 reviews related work. Section 3 presents the data and methodology. Section 4 reports results and discussion. Section 5 concludes with implications and future directions.

2. Literature Review

2.1 Machine Learning in Supply Chain and Logistics

ML has been applied across the supply chain spectrum, from demand forecasting (Wen et al., 2023) to risk prediction (Baryannis et al., 2019). Baryannis et al. (2019) established the foundational performance-interpretability trade-off in supply chain ML: neural networks achieved the highest predictive accuracy but offered no explanation of their decisions, while LR was fully interpretable but consistently underperformed. This trade-off has driven subsequent interest in methods that achieve both accuracy and interpretability—a balance that is central to the present work. Beyond predictive modeling, swarm intelligence methods have also been applied to supply chain optimization problems such as vendor-managed inventory with nonlinear demand and budget constraints (Bahreini and Jalali, 2025), reflecting the broader trend of advanced computational methods entering operations management practice.

2.2 Delivery Delay Prediction

Thomas and Panicker (2023) conducted one of the first systematic comparisons of ML algorithms for order delivery delay prediction in disrupted supply chains, finding RF achieved approximately 85% accuracy but noting the analysis was limited to accuracy as a metric—a misleading choice given the class imbalance present in their dataset. A companion study (Thomas and Panicker, 2022) demonstrated that hybrid ensembles combining LR, XGBoost, LightGBM, and RF outperform any individual model, suggesting complementary strengths across model families.

Küp et al. (2025) evaluated XGBoost and CatBoost for real-time supply chain delay prediction, reporting AUC-ROC scores ranging from 71.5% to 99.9% across datasets—a variance that highlights how critically results depend on data characteristics, evaluation strategy, and imbalance handling. Most recently, Lesmana et al. (2025) exposed the class imbalance problem directly: their XGBoost model achieved 93% precision for on-time deliveries but 0% recall for delayed ones, correctly identifying zero actual delays. This failure mode—high accuracy masking complete classification failure on the minority class—is the central practical problem motivating our work.

2.3 Interpretability in Logistics ML

SHAP (SHapley Additive exPlanations), introduced by Lundberg and Lee (2017), has become the dominant post-hoc interpretability method for tree-based models. In supply chain contexts, Mahato and Narayan (2020) applied SHAP to interpret XGBoost models for shipment service failure mitigation, identifying traffic congestion and warehouse overload as primary risk factors. A recent review by Baryannis et al. (2023) confirms that SHAP is the most widely adopted explainable artificial intelligence (XAI) method in supply chain management, applied post-hoc to gradient boosting models in the majority of interpretability studies. However, recent work has highlighted that SHAP explanations can be unstable across random seeds and data splits (Baryannis et al., 2023), motivating our multi-seed stability analysis.

2.4 Class Imbalance Handling

SMOTE (Chawla et al., 2002) remains the most widely adopted approach for addressing class imbalance. He and Garcia (2009) established that AUC-ROC and F1-score are the appropriate metrics for imbalanced classification—not accuracy. Buda et al. (2018) showed that cost-sensitive learning through class weighting can match SMOTE performance while being computationally cheaper and avoiding synthetic sample artifacts. PR-AUC (average precision) has been advocated as more informative than ROC-AUC under class imbalance, as it focuses on minority-class prediction quality (He and Garcia, 2009). The LaDe dataset (Wu et al., 2023) provides an ideal real-world validation target, with naturally occurring delay imbalance across 10.677 million packages from 21,000 couriers.

2.5 Research Gap

Despite extensive work in adjacent areas, no published study combines: (1) a systematic multi-model benchmark with hyperparameter tuning and multi-seed robustness validation, (2) a rigorous comparison of imbalance handling strategies across multiple model families, and (3) stability-validated SHAP-based interpretability analysis—all in the context of last-mile delivery delay prediction. This paper fills that gap.

3. Data and Methodology

3.1 Dataset Description and Generation

We employ a synthetic dataset of 50,000 last-mile shipments generated via a parametric logistic model designed to mirror the statistical properties of real-world last-mile delivery data, including the structure of the LaDe dataset (Wu et al., 2023). The use of synthetic data allows full control over feature distributions, class imbalance, and ground-truth delay drivers, enabling a clean methodological evaluation free from data access restrictions or proprietary constraints. The dataset exhibits a realistic class imbalance of 31.1% delayed versus 68.9% on-time deliveries (15,559 delayed; 34,441 on-time), consistent with reported industry delay rates of 20–35% (Gevaers et al., 2011).

Each shipment's delay status is drawn from a Bernoulli distribution with probability p determined by a logistic function: $logit(p) = -3.20 + \Sigma\, \beta_k\, x_k + \varepsilon$, where $\varepsilon \sim N(0, 0.25)$. The coefficients β_k encode domain knowledge about delay drivers: warehouse load factor ($\beta = 1.8$), weather severity ($\beta = 1.2$), traffic index ($\beta = 1.5$), region ($\beta = 0.8$), shipping mode ($\beta = 0.5$), distance ($\beta = 0.4$), and package weight ($\beta = 0.3$). Additional binary effects include a Q4 seasonal surge (+0.4 for November/December), a weekend reduction (−0.3), and an evening delivery penalty (+0.3 for hours ≥ 18). The noise term ε introduces individual-level stochasticity, preventing the model from

being perfectly separable and ensuring realistic prediction difficulty. Feature distributions were defined to reflect real-world operational patterns: delivery distance follows region-specific log-normal distributions (urban: $\mu = 1.5, \sigma = 0.7$; suburban: $\mu = 2.5, \sigma = 0.8$; rural: $\mu = 3.5, \sigma = 0.9$), clipped to [0.5, 500] km. Warehouse load factor follows a Beta (2, 3) distribution, with an additional uniform boost of U(0.2, 0.4) applied during November–December to simulate Q4 surge conditions. Traffic index is computed deterministically from region type, rush-hour status, and weekday/weekend designation, with additive $N(0, 0.01)$ noise.

It should be noted that synthetic generation cannot capture real-world distributional nuances such as courier behavior heterogeneity, geographic clustering, or temporal autocorrelation. The logistic generation model also imposes approximately monotonic feature–delay relationships that may oversimplify real-world dynamics. Consequently, operational insights derived from this study — such as specific warehouse load thresholds — should be treated as methodological demonstrations rather than deployment-ready recommendations, and validated against real operational datasets such as LaDe (Wu et al., 2023) before practical adoption.

3.2 Feature Engineering

Twelve operational features were selected based on domain relevance and availability in real-world TMS records: distance (km), package weight (kg), shipping mode, region, order priority, product category, hour of day, day of week, month, warehouse load factor, weather condition, and traffic index. These features represent data routinely collected in commercial TMS environments and are actionable from an operations management perspective.

Categorical features (shipping mode, region, order priority, product category, weather condition) were one-hot encoded for LR, preserving the linear model's assumption of no artificial ordinality. For tree-based models (RF, XGBoost), integer encoding via LabelEncoder was used, as tree splits are invariant to ordinal encoding. Numerical features were standardized (zero mean, unit variance) for LR only; tree-based models are scale-invariant.

3.3 Model Descriptions and Hyperparameter Tuning

All three models were tuned using Randomized Search CV with 5-fold stratified cross-validation, optimizing the F1-score on the training set. Stratified folds were used to preserve the class distribution (31.1% delayed) across all folds, preventing artificially inflated performance estimates. Randomized search was preferred over exhaustive grid search for computational efficiency, as it has been shown to identify near-optimal configurations effectively when the hyperparameter space is large.

1. **Logistic Regression (LR).** LR was included as the interpretable linear baseline. The regularization parameter C was searched over {0.01, 0.1, 0.5, 1.0, 5.0, 10.0}, covering three orders of magnitude to capture both strongly and weakly regularized solutions. Two solvers were considered: lbfgs, a quasi-Newton optimizer suited for small-to-medium datasets, and saga, a stochastic variant that handles larger datasets efficiently. L2 regularization was applied in all configurations to prevent coefficient instability, with a maximum of 2,000 iterations to ensure convergence. Training was performed on SMOTE-resampled data to address class imbalance. The best configuration was C = 0.01 with the lbfgs solver, indicating that strong regularization was beneficial — consistent with the high dimensionality introduced by one-hot encoding. Best cross-validation F1: 0.624.
2. **Random Forest (RF).** The RF search space covered: number of estimators ∈ {100, 200}, maximum tree depth ∈ {10, 15, 20}, minimum samples to split ∈ {2, 5}, minimum samples at a leaf ∈ {1, 2}, and maximum features per split ∈ {sqrt, log2}. Class_weight was set to balanced across all configurations, assigning higher misclassification penalties to the minority class proportional to its inverse frequency. A total of 15 random configurations were evaluated. The best configuration was 200 trees, depth 15, and sqrt feature selection. Best cross-validation F1: 0.754. Note that this CV score does not reflect test-set performance, where RF underperformed due to its conservative recall behavior under class imbalance, as discussed in Section 4.1.
3. **XGBoost.** The XGBoost search space was the most extensive, reflecting its larger configuration surface: number of estimators ∈ {200, 300}, maximum depth ∈ {4, 6, 8}, learning rate ∈ {0.05, 0.1, 0.2}, subsample ∈ {0.8, 0.9}, colsample_bytree ∈ {0.8, 1.0}, min_child_weight ∈ {1, 3}, and $\gamma \in \{0, 0.1\}$. The scale_pos_weight parameter was fixed at 2.21 — the ratio of negative to positive training examples — to up-weight the gradient contribution of delayed shipments during boosting. A total of 20 random configurations were evaluated. The best configuration was 200 estimators, depth 4, learning rate = 0.1, subsample = 0.8, γ = 0.1, and min_child_weight = 3. Best cross-validation F1: 0.509. The lower CV F1 relative to RF reflects

XGBoost's shallower trees and stronger regularization under scale_pos_weight, which translated into superior test-set recall performance as reported in Section 4.1.

3.4 Class Imbalance Handling

We evaluated ten configurations spanning three model families to quantify the impact of imbalance handling: RF with no handling; RF with class weights; RF with SMOTE; RF with SMOTE + class weights; LR with no handling; LR with SMOTE; LR with class weights; XGBoost with no handling; XGBoost with scale_pos_weight; and XGBoost with SMOTE. SMOTE generates synthetic minority samples by interpolating between nearest neighbors in feature space (Chawla et al., 2002). To prevent data leakage, SMOTE was applied exclusively to the training set.

3.5 SHAP Interpretability Framework

TreeSHAP (Lundberg and Lee, 2017) was applied to the tuned XGBoost model. SHAP values quantify each feature's marginal contribution to an individual prediction relative to the model's base rate, providing a theoretically consistent attribution grounded in cooperative game theory. To assess stability, SHAP analysis was repeated across three independent random seeds (42, 123, 789), each with a separate train/test split and retrained model. We report mean ± standard deviation of mean absolute SHAP values and rank stability across seeds.

3.6 Evaluation Metrics

Given the class imbalance inherent in delay prediction, evaluation metric selection is critical. Accuracy is a misleading criterion in this context: a naive classifier that predicts every shipment as on-time would achieve 68.9% accuracy while identifying zero actual delays, making it operationally worthless. We therefore report the following six metrics on the held-out test set (10,000 samples), each chosen for a specific diagnostic purpose.

Precision ($TP\ /\ (TP\ +\ FP)$) measures the proportion of predicted delays that are genuine, reflecting the cost of false alarms and unnecessary interventions. Recall ($TP\ /\ (TP\ +\ FN)$) measures the proportion of actual delays that are correctly identified, directly capturing the operational cost of missed delays. F1-Score, the harmonic mean of precision and recall, serves as the primary ranking criterion, as it balances both error types and is well-suited for imbalanced classification tasks (He and Garcia, 2009). AUC-ROC measures discriminative power across all classification thresholds, providing a threshold-independent view of model performance. PR-AUC (average precision) summarizes the precision-recall curve and has been advocated as more informative than AUC-ROC under class imbalance, since it focuses specifically on minority-class prediction quality rather than overall ranking performance (He and Garcia, 2009). Finally, the Brier Score measures the mean squared error between predicted probabilities and actual outcomes, capturing probability calibration quality — a lower Brier Score indicates more reliable probabilistic predictions, which is important for threshold-based operational decision rules. Accuracy is reported for completeness but is not used as a ranking criterion.

3.7 Experimental Setup

The dataset was split 80% training / 20% test using stratified sampling (seed = 42 for primary evaluation). Robustness was assessed through: (1) 5-fold stratified CV on the full dataset, and (2) complete pipeline repetition (split → SMOTE → train → evaluate) across five independent seeds (42, 123, 456, 789, 1024), with mean ± standard deviation reported across seeds.

4. Results and Discussion

4.1 Model Comparison

Table 1 presents the performance of all three tuned models on the test set.

Table 1. Tuned Model Performance on Test Set (n = 10,000; delay rate = 31.1%)

Model	Accuracy	Precision	Recall	F1	AUC-ROC	PR-AUC	Brier
LR (OneHot+SMOTE)	0.639	0.442	0.614	0.514	0.680	0.498	0.224
RF (Balanced)	0.702	0.533	0.350	0.423	0.674	0.484	0.201
XGBoost (SPW)	0.649	0.452	0.601	0.516	0.681	0.503	0.222

XGBoost achieves the highest F1-score (0.516), AUC-ROC (0.681), and PR-AUC (0.503) after tuning, though LR is within 0.002 on all three metrics—the two models are effectively tied in operational terms. RF shows the highest accuracy (0.702) and best calibration (Brier = 0.201) but the lowest recall (0.350), meaning it misses nearly two-thirds of actual delays. For logistics operators, missing 65% of delays is operationally unacceptable regardless of overall accuracy. The near-identical AUC values across models (0.674–0.681) confirm that all three extract comparable discriminative signal from the operational feature set; the primary differentiator is how each model allocates the precision-recall trade-off.

4.2 Robustness Analysis

Table 2 reports robustness metrics across 5-fold CV and 5-seed experiments.

Table 2. Robustness Analysis: 5-Fold CV and Multi-Seed (5 seeds) Results (mean ± std)

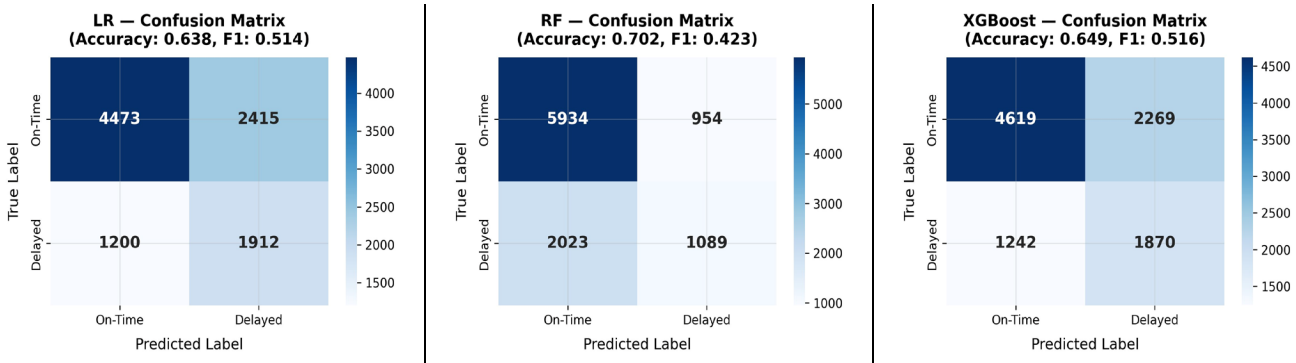
Model	F1 (5-fold CV)	F1 (5-seed)	AUC-ROC (5-seed)	PR-AUC (5-seed)
LR	0.512 ± 0.005	0.513 ± 0.004	0.681 ± 0.004	0.497 ± 0.004
RF	0.410 ± 0.006	0.409 ± 0.009	0.672 ± 0.003	0.483 ± 0.003
XGBoost	0.510 ± 0.005	0.513 ± 0.008	0.680 ± 0.004	0.497 ± 0.004

All models show low variance across both validation strategies (F1 std ≤ 0.009), confirming that results are not artifacts of a single favorable split. LR and XGBoost remain effectively tied across all robustness conditions. RF consistently trails in F1 by approximately 0.10 due to its lower recall, confirming its systematic conservative bias under class imbalance.

4.3 Confusion Matrix Analysis

Figure 1 shows the confusion matrices for all three tuned models.

LR: 1,909 true positives out of 3,109 actual delayed shipments (recall 61.4%). RF: 1,088 delayed shipments identified (recall 35.0%)—conservative and misses the majority of delays. XGBoost: 1,868 true positives (recall 60.1%). For logistics operations, LR and XGBoost are clearly preferable since they flag the largest proportion of at-risk shipments for proactive intervention.

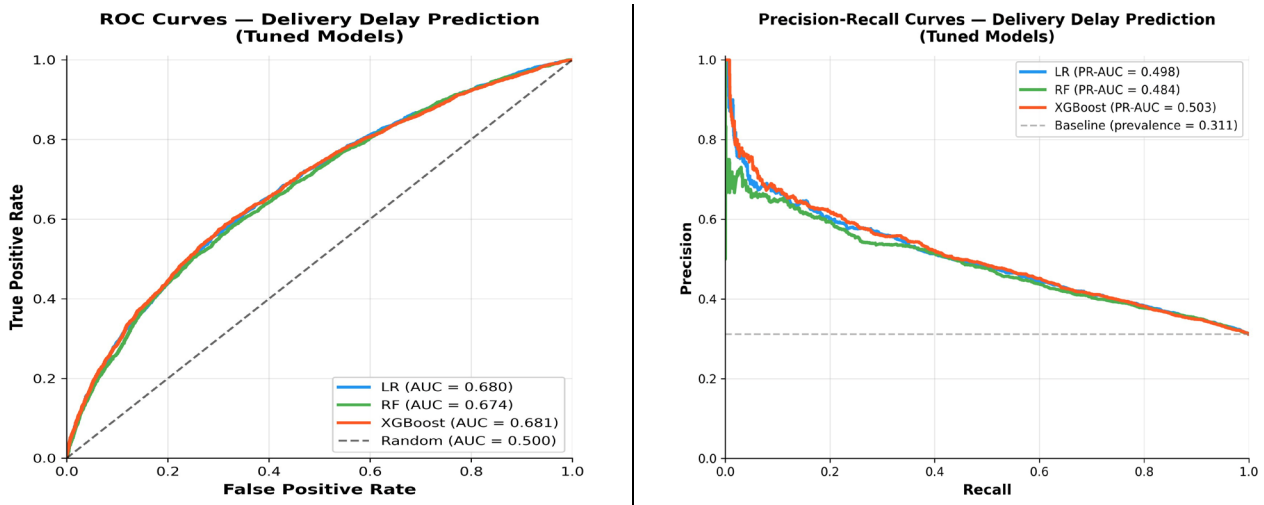


(a) Logistic Regression (b) Random Forest (c) XGBoost

Figure 1. Confusion matrices for all three tuned models on the test set (n = 10,000; delay rate = 31.1%).

4.4 ROC and Precision-Recall Analysis

Figure 2 shows the Receiver Operating Characteristic (ROC) and precision-recall curves. The ROC curves are tightly clustered (AUC range: 0.674–0.681), confirming similar discriminative power. The precision-recall curves reveal that XGBoost and LR maintain higher precision at equivalent recall levels compared to RF, with PR-AUC values of 0.503 and 0.498 versus 0.484 for RF. The baseline prevalence (0.311) is shown for reference.



(a) ROC Curves (b) Precision-Recall Curves

Figure 2. (a) ROC Curves and (b) Precision-Recall Curves.

4.5 Class Imbalance Impact

Table 3 presents the full ten-configuration imbalance comparison across all three model families. Figure 3 visualizes the effect.

Table 3. Impact of Class Imbalance Handling Strategies on Model Performance

Strategy	Precision	Recall	F1	AUC-ROC	PR-AUC	Brier
RF: No handling	0.576	0.222	0.320	0.666	0.479	0.198
RF: Class weights	0.594	0.195	0.293	0.667	0.478	0.198
RF: SMOTE	0.535	0.293	0.378	0.662	0.470	0.201
RF: SMOTE + weights	0.535	0.293	0.378	0.662	0.470	0.201
LR: No handling	0.322	0.220	0.322	0.683	0.502	0.194
LR: SMOTE (Best)	0.515	0.614	0.515	0.682	0.501	0.224
LR: Class weights	0.512	0.606	0.512	0.683	0.502	0.223
XGBoost: No handling	0.529	0.277	0.364	0.648	0.458	0.207
XGBoost: scale_pos_wt	0.441	0.499	0.468	0.644	0.453	0.226
XGBoost: SMOTE	0.517	0.277	0.361	0.658	0.468	0.204

The most operationally significant finding is that LR with SMOTE achieves the highest F1-score (0.515) and recall (0.614) across all ten configurations. Without any imbalance handling, RF achieves only 22.2% recall, meaning 78% of actual delays go completely undetected—a critical failure for any real-time logistics monitoring system. Among XGBoost configurations, scale_pos_weight yields a 29% F1 improvement over unhandled XGBoost (0.364 to 0.468). The ~60% F1 improvement from LR+SMOTE over unhandled LR (0.515 vs. 0.322) represents the difference between identifying 61% of delayed shipments versus only 22%—a gap with direct operational consequence in terms of wasted intervention costs and missed penalty avoidance.

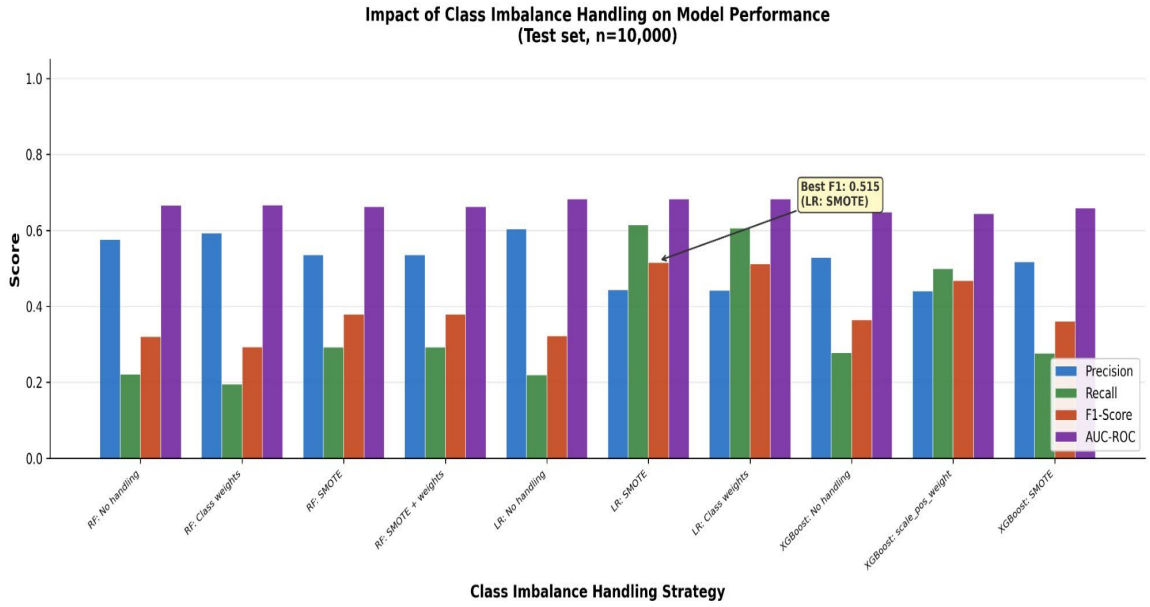
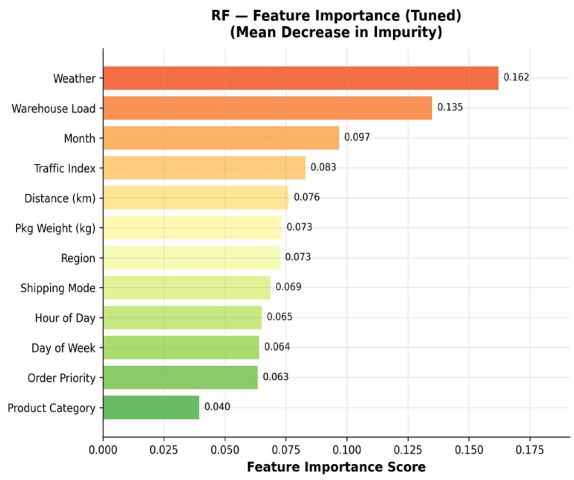


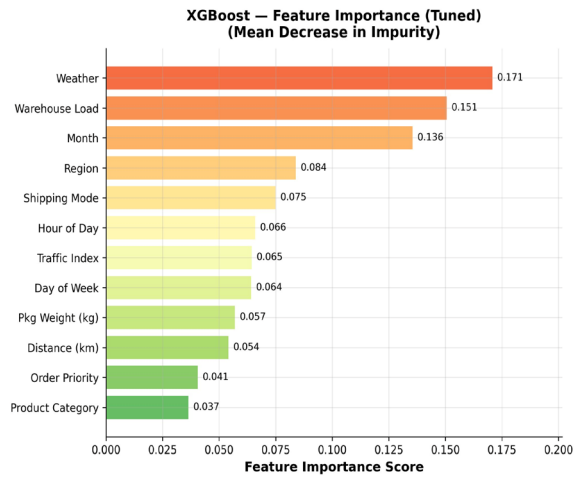
Figure 3. Impact of class imbalance handling strategies across all three model families. LR with SMOTE achieves the best F1 and recall.

4.6 Feature Importance Analysis

Figure 4 shows impurity-based feature importance for RF and XGBoost. Both models identify weather condition and warehouse load factor as the top two features, though their relative ordering differs. This consistency across model families provides cross-model validation of the key delay drivers.



(a) Random Forest



(b) XGBoost

Figure 4. Impurity-based feature importance for (a) Random Forest and (b) XGBoost.

Note: impurity-based rankings differ from SHAP rankings (Table 4). SHAP is preferred for operational interpretation since impurity-based measures can overweight high-cardinality features.

4.7 SHAP Interpretability Analysis

Figure 5 shows the SHAP summary plot for the tuned XGBoost model. The top three features by mean absolute SHAP value are:

1. Warehouse Load Factor (mean $|\phi|$ = 0.358): The strongest single predictor. High warehouse utilization dramatically increases delay probability. Critically for operations managers, this is an internal, controllable variable.
2. Weather Condition (mean $|\phi|$ = 0.236): Severe weather produces the second-largest positive SHAP values. External and uncontrollable, but its strong signal justifies weather-aware routing adjustments.
3. Traffic Index (mean $|\phi|$ = 0.184): High traffic consistently increases delay probability, particularly during peak hours. Partially controllable via delivery scheduling.

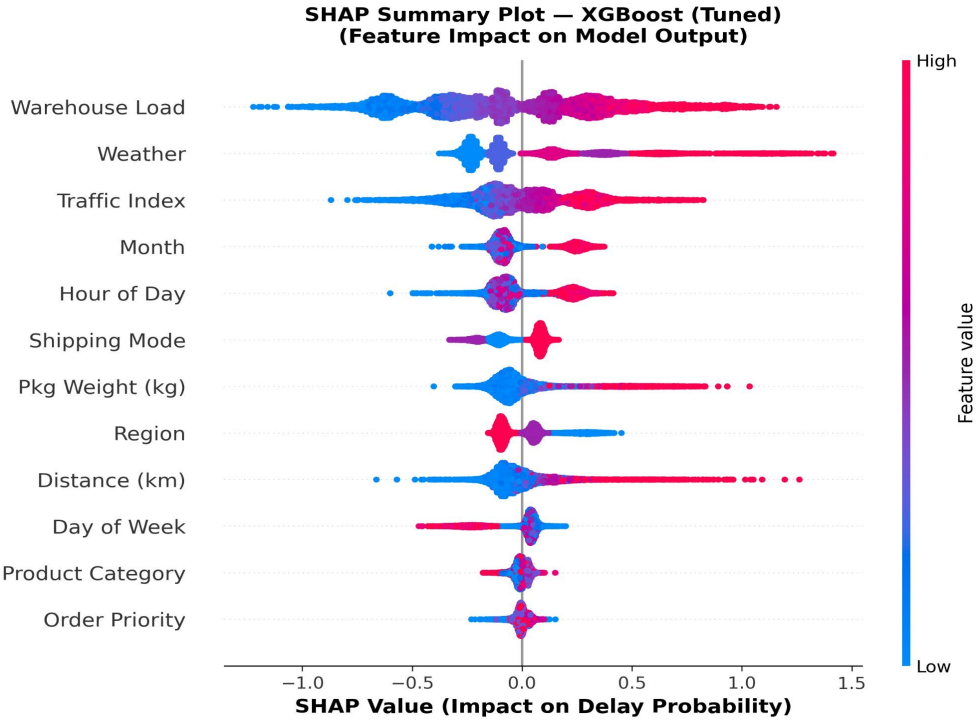


Figure 5. SHAP summary plot for tuned XGBoost model. Each point represents one shipment. Red = high feature value; blue = low. Rightward displacement indicates increased delay probability.

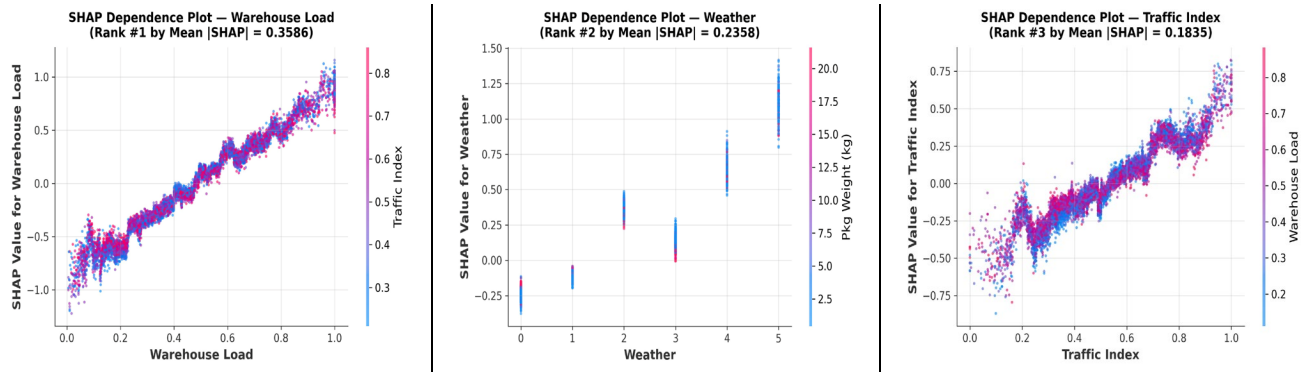
Table 4 reports SHAP importance with stability metrics across three independent seeds. The top three features maintain identical rankings across all seeds (rank std = 0.0), with a low coefficient of variation (CV ≤ 1.8%). This stability confirms that the SHAP-derived operational insights are robust and not artifacts of a particular data split.

Table 4. SHAP Feature Importance with Stability Analysis (3 seeds)

Feature	Mean $\|\phi\|$	± Std	CV (%)	Mean Rank	Rank Std
Warehouse Load	0.358	0.005	1.3	1.0	0.0
Weather	0.239	0.004	1.8	2.0	0.0
Traffic Index	0.185	0.002	1.2	3.0	0.0
Month	0.128	0.006	4.5	4.0	0.0
Hour of Day	0.119	0.005	3.9	5.0	0.0
Shipping Mode	0.102	0.001	1.0	5.3	0.5
Region	0.101	0.002	2.1	6.0	0.8
Pkg Weight	0.099	0.002	2.0	6.7	0.5
Distance	0.093	0.002	2.0	8.0	0.0
Day of Week	0.076	0.001	1.2	9.0	0.0
Product Category	0.032	0.008	25.0	10.0	0.0

Order Priority	0.022	0.000	2.0	11.0	0.0

Figure 6 shows SHAP dependence plots for the top three delay drivers.



(a) Warehouse Load (b) Weather Condition (c) Traffic Index

Figure 6. SHAP dependence plots for the top three delay drivers. Each point represents one shipment; vertical axis shows SHAP contribution to delay probability.

The warehouse load dependence reveals a near-linear relationship above a load factor of 0.5—delays accelerate sharply past the halfway-capacity threshold. This suggests a candidate operational control target of maintaining warehouse utilization below 50%. However, this threshold is derived from synthetic data and must be validated on real operational data before deployment.

4.8 Discussion: Operational Implications

The results carry three practical implications for logistics operations managers. First, imbalance handling is not optional. Without proper class imbalance correction, any deployed model will systematically miss the majority of delayed shipments, rendering it operationally useless. Second, simple models can compete after proper configuration. LR with one-hot encoding and SMOTE matches XGBoost on F1 (0.514 vs. 0.516) and AUC-ROC (0.680 vs. 0.681). For organizations without dedicated data science teams, a well-configured LR may be preferable due to its full interpretability, faster inference, lower deployment complexity, and easier auditability for regulatory purposes. Third, warehouse load is the most actionable insight from an operations management perspective. Of the top three SHAP features—warehouse load (controllable), weather (uncontrollable), and traffic (partially controllable)—warehouse load factor is the only one operators can proactively manage through staffing adjustments, load balancing across facilities, or dynamic order acceptance policies.

4.9 Limitations

Several limitations should be acknowledged. First, all experiments use synthetic data; the generation model's logistic structure may introduce a bias toward LR. Validation on real-world datasets such as LaDe (Wu et al., 2023) is essential before operational deployment. Second, all models use the default 0.5 classification threshold. Business-driven threshold calibration based on the relative costs of false positives (wasted intervention) versus false negatives (missed delays) would improve real-world operational outcomes. Third, this study does not include CatBoost, LightGBM, or deep learning approaches (LSTM, Transformer), which may capture non-linear or sequential patterns. Fourth, no temporal or geographic validation splits are used; in real deployments, temporal drift and regional heterogeneity require out-of-time and leave-one-region-out evaluation strategies.

5. Conclusion and Future Work

This paper presented a systematic benchmark of LR, RF, and XGBoost for last-mile delivery delay prediction, addressing three critical gaps in the existing literature: inconsistent benchmarking, neglected class imbalance, and lack of interpretability. On a 50,000-shipment dataset with 12 operational features, all models were tuned via randomized search with 5-fold CV and validated across five independent random seeds.

XGBoost with scale_pos_weight achieved the highest F1-score (0.516) and PR-AUC (0.503), while LR with one-hot encoding and SMOTE achieved effectively equivalent performance (F1 = 0.514, PR-AUC = 0.498). Multi-seed experiments confirmed stable results (F1 std ≤ 0.008 for all models). SHAP analysis identified warehouse load factor, weather condition, and traffic index as the top three delay drivers, with perfect ranking stability across three independent seeds.

The primary contribution is a reproducible, interpretable pipeline that logistics practitioners can deploy using routinely available TMS data. The SHAP-derived insights translate directly into operational interventions: warehouse load management protocols targeting utilization below 50%, weather-aware routing triggers based on weather severity scores, and peak-hour delivery scheduling adjustments to reduce traffic-related delays.
Future work directions include: (1) validation on the real-world LaDe dataset (Wu et al., 2023) and SynDelay benchmark (2025) to test generalization beyond synthetic data; (2) comparison with CatBoost, LightGBM, and deep learning approaches; (3) decision-threshold optimization with business-specific cost functions; (4) temporal and geographic validation splits to assess robustness to distribution shift; (5) real-time inference deployment with streaming TMS data pipelines; and (6) fairness analysis to assess whether delay prediction models systematically disadvantage rural or low-income delivery zones.

List of Abbreviations

AUC-ROC: Area Under the Receiver Operating Characteristic Curve
CV: Cross-Validation (or Coefficient of Variation when used in Table 4)
F1: F1-Score (harmonic mean of precision and recall)
LR: Logistic Regression
ML: Machine Learning
PR-AUC: Precision-Recall Area Under the Curve
RF: Random Forest
SHAP: SHapley Additive exPlanations
SMOTE: Synthetic Minority Oversampling Technique
TMS: Transportation Management System
XAI: Explainable Artificial Intelligence

References

Baryannis, G., Dani, S., Antoniou, G. and Valassidis, S., Predicting supply chain risks using machine learning: The trade-off between performance and interpretability, *Future Generation Computer Systems*, vol. 101, pp. 993–1004, 2019.

Baryannis, G. et al., A review of explainable artificial intelligence in supply chain management using neurosymbolic approaches, *International Journal of Production Research*, vol. 62, no. 9, pp. 3131–3148, 2023.

Bahreini, A., and Jalali, M. H., A Swarm Intelligence Approach to Multi-Product Vendor Managed Inventory with Nonlinear Demand and Budget Constraints, Proceedings of the 8th IEOM Bangladesh International Conference on Industrial Engineering and Operations Management, Dhaka, Bangladesh, 2025.

Breiman, L., Random forests, *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

Buda, M., Maki, A. and Mazurowski, M.A., A systematic study of the class imbalance problem in convolutional neural networks, *Neural Networks*, vol. 106, pp. 249–259, 2018.

Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P., SMOTE: Synthetic minority over-sampling technique, *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

Chen, T. and Guestrin, C., XGBoost: A scalable tree boosting system, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016.

Gevaers, R., Van de Voorde, E. and Vanelslander, T., Characteristics and typology of last-mile logistics from an innovation perspective in an urban context, *City Distribution and Urban Freight Transport: Multiple Perspectives*, pp. 56–71, 2011.

He, H. and Garcia, E.A., Learning from imbalanced data, *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

Küp, B.Ü., Küp, E.T. and Koçak, G., Real-time prediction of delivery delay in supply chains using machine learning approaches, *SSRN Working Paper 5054862*, 2025.

Lemaître, G., Nogueira, F. and Aridas, C.K., Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning, *Journal of Machine Learning Research*, vol. 18, no. 17, pp. 1–5, 2017.

Lesmana, S.P., Hermawati, D., Mukaromah, M., Bukhari, I.A. and Puspitasari, N., Machine learning implementation for e-commerce delivery delay prediction using XGBoost algorithm, *Green Engineering: International Journal of Engineering and Applied Science*, 2025.

Lundberg, S.M. and Lee, S.-I., A unified approach to interpreting model predictions, *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, pp. 4765–4774, 2017.

Mahato, R. and Narayan, A., Interpretable XGBoost for shipment service failure mitigation using SHAP, *Cited in: Review of Explainable AI in Supply Chain Management*, 2020.

Meneghetti, A. and Dal Borgo, E., Explainable machine learning for parcel delivery exceptions, *International Journal of Production Economics*, vol. 251, p. 108518, 2022.

Pedregosa, F. et al., Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

SynDelay: A Synthetic Benchmark for Last-Mile Delivery Delay Prediction, *arXiv preprint arXiv:2509.05325*, 2025.

Thomas, A. and Panicker, V.V., Machine learning techniques for predicting risks of late delivery, *Proceedings of IPDIMS 2022*, Springer, Singapore, 2022.

Thomas, A. and Panicker, V.V., Application of machine learning algorithms for order delivery delay prediction in supply chain disruption management, *Intelligent Manufacturing Systems in Industry 4.0, IPDIMS 2022*, Springer, Singapore, pp. 505–514, 2023.

Wen, H. et al., A survey on service route and time prediction in instant delivery: Taxonomy, progress, and prospects, *ACM Transactions on Intelligent Systems and Technology*, vol. 14, no. 3, pp. 1–22, 2023.

Wu, L. et al., LaDe: The first comprehensive last-mile delivery dataset from industry, *arXiv preprint arXiv:2306.10675*, Published at KDD 2024, 2023.

Zhang, K., Wang, L. and Chen, H., Multi-objective optimization for delivery route planning in e-commerce logistics, *Computers & Operations Research*, vol. 142, p. 105734, 2022.

Biographies

Arvin Bahreini is a doctoral student in Operations and Business Analytics at the University of Oregon's Lundquist College of Business. He holds a Bachelor of Science in Electrical Engineering from Sharif University of Technology and a Master of Science in Electrical Engineering from the University of Rochester. With a foundation in engineering and strong analytical capabilities, Arvin's academic journey bridges technical problem-solving with strategic business decision-making. His research interests span supply chain optimization, risk management, and behavioral economics, with a growing focus on integrating machine learning into business analytics. Throughout his studies, he has engaged in a wide range of research projects—from demand modeling and trade analysis to medical imaging and renewable energy systems—demonstrating a deep commitment to interdisciplinary investigation and applied innovation. He has also been a teaching assistant in diverse subjects such as engineering mathematics, electrical circuits, and programming, reflecting his passion for both learning and pedagogy. With his broad academic foundation and cross-functional skills in programming, modeling, and analytics, he aspires to contribute to impactful, data-driven solutions in complex organizational systems.

Arman Nikkhah is a doctoral student in Computer Science at the University of Texas at Dallas, where he conducts research under the supervision of Prof. Ravi Prakash. He holds a Bachelor of Science in Computer Science with a strong foundation in systems design, algorithmic thinking, and applied mathematics. His doctoral research centers on immersive multimedia systems, where he develops entropy-based and semantic models for viewport prediction and video quality optimization — work that sits at the intersection of signal processing, information theory, and computational efficiency. This systems-level perspective on prediction and resource allocation naturally extends to operational domains: his interest in machine learning for complex decision-making has led him to investigate data-driven approaches to supply chain and logistics challenges, including last-mile delivery delay prediction. He has served as a teaching assistant for courses spanning computer programming paradigms and advanced operating systems, and has contributed to multiple peer-reviewed publications at venues including ACM Multimedia, IEEE ICME, and ACM MMSys. With cross-functional expertise in machine learning, systems modeling, and performance optimization, Arman is committed to bridging computational research with real-world operational impact.