

Explainable Cloud-Native Payment Fraud Detection Using SHAP-uided Feature Adaptation

Sanne de Vries, Felipe Rocha*

Informatics Institute, University of Amsterdam, Netherlands

* Corresponding author: feliparochanh@gmail.com

Abstract: Real-time payment fraud detection in cloud-native environments must reconcile three competing pressures: predictive accuracy, strict tail-latency budgets, and growing regulatory demands for explainability. Prior work has mapped the accuracy-latency-footprint frontier for CPU-friendly classifiers but has not treated explainability, nor the runtime cost of the feature pipeline itself, as first-class design concerns. We introduce SHAP-Guided Feature Adaptation (SGFA), a teacher-student method that (1) trains a random-forest teacher on all features, (2) computes exact TreeSHAP attributions on a validation window to obtain a faithful global feature ranking, (3) selects a compact feature subset through a transparent rule—the smallest number of features reaching ninety percent of the teacher’s best validation PR-AUC—and (4) retrains a lightweight student on that subset. On the public ULB credit-card dataset (284,807 transactions, 0.173% fraud) under a chronological split and a precision-floor operating point, SGFA shrinks the feature set from thirty to eight (a 73% reduction) while a logistic-regression student retains 96.5% of the full-feature PR-AUC at a p99 latency of 321 microseconds and a 1.3 KB memory footprint; fifteen features fully recover full-feature accuracy. We further show that the SHAP ranking is competitive with the strongest filter baseline while additionally yielding faithful local and global explanations at no extra modeling cost, that it passes an AOPC faithfulness test (a 2.4x improvement over random ablation), and that the stability of attributions across time windows provides a practical concept-drift signal. SGFA thus demonstrates that explainability, far from being a tax on accuracy or latency, can actively improve a cloud-native fraud-scoring pipeline.

Keywords: Fraud detection; Explainable artificial intelligence; SHAP; Feature selection; Cloud-native systems; Model interpretability; Concept drift; Real-time inference; Imbalanced classification.

1. Introduction

Digital payment volumes and the fraud losses that accompany them continue to grow, and modern payment platforms increasingly score every transaction inline—within a cloud-native microservice architecture, before the transaction is authorized [1], [2]. This deployment setting imposes constraints that are unusual for a machine-learning classifier. Scoring sits on the critical path of the payment, so the latency that matters is not the average but the tail: a service that is fast on the median but occasionally slow will, at scale, delay or degrade a large number of legitimate payments [3]. At the same time, fraud is extremely rare—often well below one percent of transactions—so accuracy must be measured with metrics suited to severe class imbalance rather than raw error rate [4], [5].

Recent work by Zi [2] characterized this regime carefully, benchmarking a family of CPU-friendly classifiers on a public credit-card dataset under a chronological split and a precision-floor operating rule, and mapping the resulting trade-offs among predictive quality, tail latency, and memory footprint. That study found, among other things, that a well-regularized logistic regression sits on a favorable region of the accuracy-latency frontier—sub-millisecond and only a few kilobytes—whereas a random forest achieves the best ranking quality but at roughly an order of magnitude more latency and several orders of magnitude more memory. These findings provide a valuable operating-point-aware baseline for production fraud scoring.

What that line of work does not address, and what has become increasingly important, is explainability. Three forces push explanations to the foreground. First, regulation: data-

protection regimes and financial-supervision rules increasingly expect that automated decisions affecting customers can be explained [6], [7]. Second, operations: fraud analysts who review flagged transactions need to know why a transaction was flagged in order to act efficiently. Third, and less obviously, the feature pipeline is itself a runtime cost and a source of operational risk. Every feature consumed by the model must be computed, fetched, or cached on the critical path, and every feature is a potential point of failure and a surface for concept drift. Latency-focused studies typically hold the feature set fixed and do not ask which features are actually earning their keep.

We argue that these two concerns—explainability and feature parsimony—are naturally addressed together, and that the tool that connects them is feature attribution. Shapley-value explanations, and in particular the exact and efficient TreeSHAP algorithm for tree ensembles [8], [9], assign each feature a contribution to each prediction with strong theoretical grounding. Such attributions are normally used only for post-hoc reporting; we instead use them as a design lever.

This paper makes the following contributions. (i) We propose SHAP-Guided Feature Adaptation (SGFA), a teacher-student procedure that uses TreeSHAP attributions from a random-forest teacher both to rank features and, via a transparent selection rule, to choose a compact subset on which a lightweight student is retrained. (ii) On the public ULB credit-card dataset we reproduce the operating-point-aware protocol of prior work and then show that eight of thirty features—a 73% reduction—retain between ninety-six and ninety-eight percent of full-feature PR-AUC, while fifteen features match it exactly, at reduced latency and footprint. (iii)

We give an honest comparison of SHAP against filter and embedded feature-selection criteria, finding that SHAP is competitive with the strongest of them on raw predictive quality while uniquely providing faithful local and global explanations at no additional modeling cost, and we verify attribution faithfulness with an ablation (AOPC) test. (iv) We show that the stability of SHAP attributions across successive time windows is a usable concept-drift diagnostic. All experiments use a public dataset and released code.

Section II reviews related work. Section III defines the SGFA method. Section IV describes the experimental setup, Section V presents results, Section VI discusses implications and limitations, and Section VII concludes.

2. Related Work

2.1. Cloud-Native Real-Time Fraud Systems

Production fraud detection has moved from nightly batch scoring toward streaming, always-on services built from microservices [1]. Frameworks such as SCARFF integrate streaming platforms with learning to score transactions at scale [10], and streaming active-learning strategies address the reality that fraud labels arrive with delay and human review capacity is limited [11], [12]. Because scoring is inline, tail latency is a primary concern; the general lesson that tail behavior dominates user-visible performance in large fan-out services applies directly [3]. Zi [2] made these constraints explicit for fraud classifiers by reporting p99 latency and serialized model size alongside predictive metrics; we adopt that operating-point-aware methodology and extend it with an explainability dimension. Treating models as long-lived services also exposes the hidden maintenance cost of machine-learning systems, including data dependencies and unused features [13], which motivates our attention to feature parsimony.

2.2. Learning with Extreme Class Imbalance

Credit-card fraud is a canonical imbalanced-learning problem. Comparative studies established strong tabular baselines and appropriate evaluation practice [4], and a body of work from Dal Pozzolo and colleagues analyzed realistic modeling choices, undersampling and probability calibration, and the effect of delayed labels [5], [14], [15]. Because the positive class is rare, threshold-free ranking metrics computed from the precision-recall curve are more informative than ROC or accuracy [16], and resampling techniques such as SMOTE are common [17]. Feature engineering—especially aggregation of transaction history—has repeatedly been shown to help [18], and sequence models and generative augmentation have also been explored [19], [20]. The strongest tabular models in this space are typically tree ensembles [21] and gradient boosting [22], [23], with linear models and stochastic-gradient variants offering cheaper alternatives [24]; we build on the same model families, implemented in scikit-learn [25].

2.3. Explainable AI and Feature Attribution

The tension between predictive power and interpretability is long-standing [26], [27], [28], and some authors argue for inherently interpretable models in high-stakes settings [29]. Among post-hoc methods, local surrogate explanations [30] and Shapley-value attributions [31], [32], [8] are the most widely used; TreeSHAP computes exact Shapley values for tree ensembles efficiently and aggregates them into global

importance [9], and practical guidance for such methods is well documented [33]. Explanations are not without pitfalls—post-hoc explainers can be manipulated by adversarially constructed models [34]—which is one reason we rely on the global, aggregate use of attributions rather than only per-instance explanations. In the fraud domain specifically, explainability has been studied both as an operational requirement for analysts and as an accuracy-interpretability trade-off [7], [35]. What is largely missing is the use of attribution as an active design lever—to shrink the model and to monitor it—rather than as a reporting layer bolted on after the fact. SGFA targets exactly this gap.

2.4. Feature Selection and Concept Drift

Feature selection is classically divided into filter, wrapper, and embedded methods. SGFA is closest in spirit to an embedded, attribution-guided selector: it uses a model’s own faithful attributions to rank features, then validates a cut point empirically. Separately, non-stationarity—concept drift—is a central challenge for deployed fraud models, and surveys catalog both detection and adaptation strategies [36], [37]. We connect these two threads by observing that the stability of a feature-attribution ranking across time windows is itself a lightweight drift signal, complementary to performance- or distribution-based detectors.

3. Method: SHAP-Guided Feature Adaptation

We now define SHAP-Guided Feature Adaptation. Fig. 1 gives an overview of the pipeline.

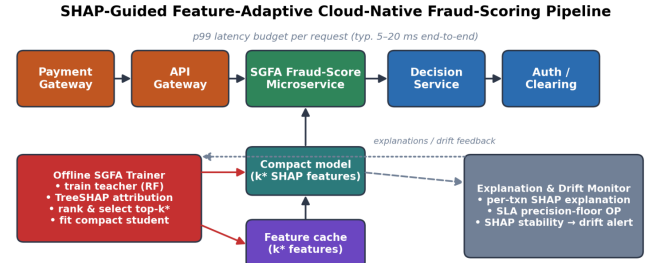


Fig. 1 The SGFA pipeline. An offline stage trains a random-forest teacher on all features and explains it with TreeSHAP on a validation window to obtain a faithful global feature ranking; a transparent rule then selects the smallest feature count reaching a target fraction of the teacher’s validation quality, and a lightweight student is retrained on that subset for online, inline scoring. The same attributions drive both feature selection and drift monitoring.

3.1. Problem Setup and Notation

Let a labeled dataset $D = \{(x, y)\}$ consist of transactions x with M candidate features and binary labels y (fraud or genuine), with fraud extremely rare. Respecting the temporal nature of transactions, we order D by time and split it chronologically into training, validation, and test windows; no shuffling is performed, so the validation and test windows always lie in the future relative to training, which avoids optimistic leakage. Because the positive class is rare, we treat the area under the precision-recall curve (PR-AUC) as the primary quality metric and report ROC-AUC as a secondary one [16]. For a deployable operating point we follow a precision-floor rule: on the validation window we choose the smallest decision threshold whose precision is at least a fixed

floor, and we then freeze that threshold for the test window, reporting precision, recall, F1, and false-positive count there. Deployment cost is summarized by single-transaction tail latency (p99, in microseconds) and serialized model size (in kilobytes).

3.2. Overview

SGFA has four stages: (1) train a high-capacity teacher on all M features; (2) explain the teacher with TreeSHAP on the validation window to obtain a faithful global feature ranking; (3) select the smallest feature count k^* that reaches a target fraction of the teacher’s best validation quality; and (4) retrain a lightweight student on those k^* features for deployment. Stages 1 and 2 are performed once, offline; only the student runs on the critical path.

3.3. Teacher Training and TreeSHAP Attribution

The teacher is a random forest [21], chosen because it attains the best ranking quality in this regime and because TreeSHAP yields exact Shapley values for it in low-order polynomial time [9]. TreeSHAP produces, for each transaction x and feature i , a contribution such that the model output is recovered by the additive form

$$g(z') = \varphi_0 + \sum \varphi_i z'_i \quad (1)$$

where the first term is the expected model output and z' indicates feature presence. The contributions are the Shapley values

$$\varphi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|! (|F| - |S| - 1)!}{|F|!} \cdot [f(S \cup \{i\}) - f(S)] \quad (2)$$

from cooperative game theory [31], [32], which uniquely satisfy local accuracy, missingness, and consistency [8]. We aggregate local attributions into a global importance for feature j by averaging absolute contributions over an evaluation set V drawn from the validation window,

$$I_j = (1/|V|) \sum_{x \in V} |\varphi_j(x)| \quad (3)$$

To bound cost while preserving the rare positive signal, V comprises all fraud cases in the validation window together with a random sample of genuine cases. Ranking features by their global importance gives the SGFA importance order.

3.4. Transparent Selection of the Feature Count

Rather than fixing the number of features by hand, we choose it with a transparent, reproducible rule. Let the validation PR-AUC of a student trained on the top- k features in SGFA order be denoted $\text{PRAUC}_{\text{val}}(k)$. We define

$$k^* = \min \{ k : \text{PRAUC}_{\text{val}}(k) \geq \tau \cdot \max \text{PRAUC}_{\text{val}} \}, \quad \tau = 0.9 \quad (4)$$

that is, the smallest feature count reaching ninety percent of the best achievable validation PR-AUC along the curve. The ratio exposes the accuracy-parsimony trade-off as a single, auditable knob.

3.5. Student Retraining

On the selected top- k^* features we retrain a lightweight student. We consider two: a logistic-regression student, which is linear, tiny, and sub-millisecond, and thus attractive when footprint and latency dominate [24]; and a compact random-forest student, which trades size for a little more ranking quality. The student, not the teacher, is deployed. Algorithm 1

summarizes the full procedure.

Algorithm 1 SHAP-Guided Feature Adaptation (SGFA)

Input: chrono split (train, val, test); features F ;
precision floor p_0 ; ratio τ

Output: student S^* on k^* features

```

1: T <- RandomForest teacher on train[F]
2: Phi <- TreeSHAP(T, V) # V: val eval set
3: for j in F: I_j <- mean_{x in V} |Phi_j(x)|
4: pi <- features sorted by I_j (descending)
5: for k in grid:
    s_k <- student(top-k of pi) on train
    a_k <- PR-AUC(s_k, val)
6: A* <- max_k a_k
7: k* <- min{ k : a_k >= tau * A* }
8: S* <- student(top-k* of pi) on train
9: th <- val threshold s.t. precision >= p0 (freeze)
10: return S*, top-k* features, th

```

3.6. Faithfulness Check

Feature selection is only trustworthy if the attributions that drive it are faithful to the model. We test this with an ablation curve: removing features from most to least important (in SGFA order) should degrade the model faster than removing them in random order. We summarize the mean output drop after removing the top- k features as an aggregated measure (AOPC),

$$\text{AOPC}(K) = (1/K) \sum_{k=1}^K [f(x) - f(x_{\setminus \text{top-}k})] \quad (5)$$

and compare the area under the SGFA-order curve against the random-order curve; a large ratio indicates faithful attributions.

3.7. Drift Diagnosis via Attribution Stability

Finally, we reuse the attribution machinery for monitoring. We compute the global importance vector on different time windows and compare successive windows in two ways: the Spearman rank correlation between importance vectors, and the Jaccard overlap of the top- k feature sets,

$$\rho(I^{(a)}, I^{(b)}), \quad J = |T_k^{(a)} \cap T_k^{(b)}| / |T_k^{(a)} \cup T_k^{(b)}| \quad (6)$$

A sharp drop in either quantity signals that the features the model relies on are shifting—an early, model-centric indicator of concept drift that complements performance- and distribution-based detectors [36], [37].

In deployment terms, the expensive steps—teacher training and TreeSHAP—occur offline and can be rerun periodically; the online path computes only k^* features and evaluates a small student, so SGFA reduces both the model and the feature pipeline that feeds it.

4. Experimental Setup

4.1. Dataset

We use the public ULB credit-card fraud dataset, a widely used benchmark of 284,807 real transactions collected over two days by European cardholders, of which 492 are fraudulent—a base rate of 0.173% [5], [14]. For confidentiality the features are the output of a PCA transformation, denoted V1 through V28, together with the elapsed Time and the transaction Amount, giving thirty candidate features. Using this dataset keeps our protocol and results directly comparable to the prior operating-point-aware benchmark on the same data.

4.2. Split and Operating Point

We sort transactions by Time and split them chronologically into 60/20/20 training, validation, and test windows. The fraud rate differs across windows (0.211%, 0.100%, and 0.132% for train, validation, and test, respectively), reflecting genuine temporal non-stationarity and making the task realistically hard. The precision floor is set to 0.10; the threshold achieving it is selected on validation and frozen for test. We report PR-AUC and ROC-AUC (threshold-free) and, at the frozen operating point, precision, recall, F1, and false positives.

4.3. Latency and Size Protocol

All models run on CPU only, with no GPU. Single-transaction latency is measured by scoring one row at a time—the inline-scoring scenario—over 500 sampled transactions after 50 warmup calls, repeated three times, from which we report median, p95, and p99 in microseconds and throughput in rows per second. Model size is the serialized on-disk footprint in kilobytes. Software is Python 3.12 with scikit-learn 1.8 [25] and the shap library for TreeSHAP.

4.4. SGFA Configuration and Baselines

The teacher is a random forest; TreeSHAP is computed on a validation evaluation set consisting of all validation-window fraud cases plus 4,000 randomly sampled genuine cases. Students are a logistic regression and a compact random forest. The candidate grid for the feature count is {3, 5, 8, 10, 12, 15, 20, 25, 30}, and the selection ratio is 0.9. To assess whether SHAP is a good selection criterion, we compare it against three alternatives at matched feature counts: mutual information (a filter), the magnitude of standardized logistic-regression coefficients (an embedded criterion), and random subsets (averaged over repeats) as a floor. All randomness uses a fixed seed.

5. Results

5.1. Reproducing the Operating-Point Baseline

We first confirm that our pipeline reproduces prior results. Table I compares our test-set metrics for three representative models against the values reported by Zi [2]. The agreement is close: logistic regression reaches a PR-AUC of 0.744 (versus 0.744 reported), the random forest 0.801 (versus 0.812), and histogram gradient boosting 0.344 (versus 0.344). Minor differences at the operating-point level are expected, since precision-floor thresholding is sensitive to small probability shifts, but the ranking metrics align, validating the setup we build on.

Table 1. Reproducing the operating-point baseline (test set)

Model	ROC-AUC	PR-AUC	PR-AUC (Zi [2])
Logistic Regression	0.982	0.744	0.744
Random Forest	0.960	0.801	0.812
Histogram Grad. Boosting	0.730	0.344	0.344

5.2. Global Attributions and the Selected Features

Fig. 2 shows the global SHAP importance of the thirty features (left) and the distribution of per-transaction contributions (right). A small number of PCA components

dominate: the top eight features in SGFA order are V12, V11, V4, V14, V10, V17, V3, and V16. These coincide with components that are independently known to be discriminative for this dataset, a reassuring sanity check that the attributions capture genuine signal rather than noise.

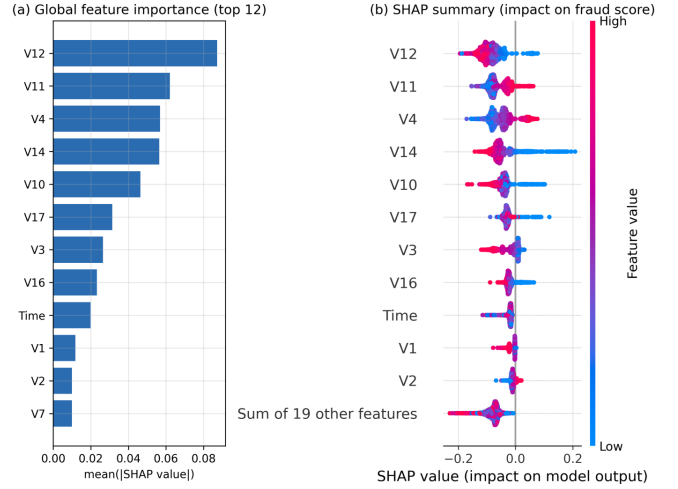


Fig. 2 Global TreeSHAP attribution for the random-forest teacher. Left: mean absolute SHAP value per feature (global importance), which defines the SGFA ranking. Right: per-transaction SHAP contributions (beeswarm), showing both the magnitude and the direction in which each feature moves the fraud score. A handful of components account for most of the attributed signal.

5.3. How Many Features Are Needed?

Fig. 3 plots student PR-AUC as a function of the number of top features retained, for both the logistic-regression and random-forest students, with horizontal lines marking the full-feature reference. Quality rises steeply and then saturates; applying the 0.9 rule to the validation curve selects $k^* = 8$. Table II reports the resulting deployed models on the test set.

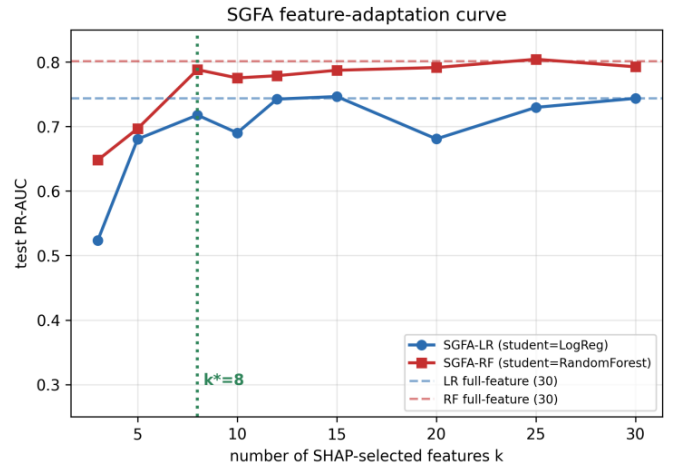


Fig. 3 Test PR-AUC of the SGFA students versus the number of retained top features. Both students recover most of their full-feature quality (dashed lines) with far fewer features; the transparent 0.9 rule selects $k^* = 8$ (marked).

The logistic-regression student on eight features retains 96.5% of the full-feature PR-AUC (0.718 versus 0.744) while shrinking to 1.33 KB and a p99 of 321 microseconds—faster and smaller than the already-lightweight full logistic model. Extending to fifteen features slightly exceeds full-feature PR-AUC (0.746). The compact random-forest student on eight features retains 98.4% of its full-feature PR-AUC (0.788 versus 0.801). In short, roughly a quarter of the features carry

almost all of the signal.

Table 2. Deployed models: full-feature vs. Sgfa (test set)

Model	# Feat	ROC-AUC	PR-AUC	% of full	p99 (μ s)	Size (KB)
LR (full)	30	0.982	0.744	100.0	450	2.02
SGFA-LR ($k^* = 8$)	8	0.981	0.718	96.5	321	1.33
SGFA-LR (top-15)	15	0.980	0.746	100.4	273	1.55
RF (full)	30	0.960	0.801	100.0	4851	797
SGFA-RF ($k^* = 8$)	8	0.963	0.788	98.4	4608	811

5.4. Latency and Footprint

Fig. 4 places these models on the latency-quality plane. The logistic-regression students occupy a favorable region: the eight-feature student pushes p99 down to 321 microseconds at a footprint of about one kilobyte, extending the favorable frontier identified in prior work toward lower cost. The random-forest students remain expensive—several milliseconds of p99 and hundreds of kilobytes—because that cost is intrinsic to tree ensembles and is not removed by pruning features; reducing the input features barely changes the serialized forest size. This is an honest limitation: where footprint is critical, the linear student is the right choice, and SGFA makes that student nearly as accurate as the full model.

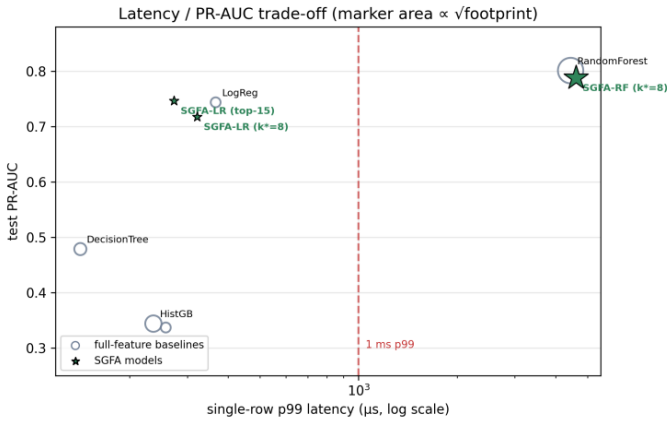


Fig. 4 Tail-latency versus PR-AUC. The SGFA logistic-regression students sit on a favorable region of the frontier (low latency, small footprint); random-forest variants retain higher PR-AUC but at markedly higher latency and size.

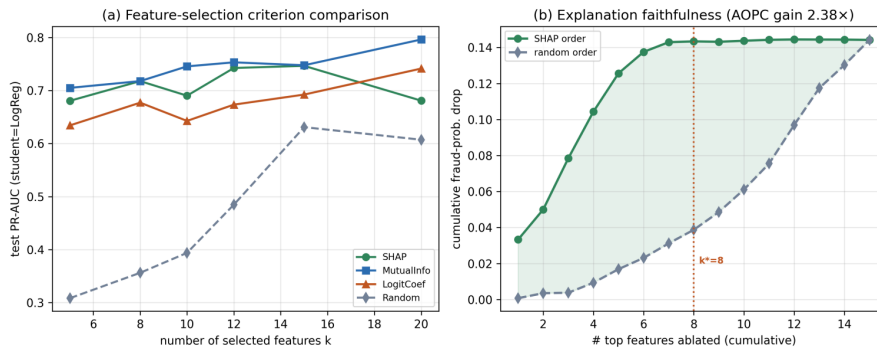


Fig. 5 (a) Feature-selection criteria compared by test PR-AUC across the feature-count grid: SHAP is competitive with mutual information and far above random selection. (b) Faithfulness: ablating features in SHAP order degrades the model much faster than random order (AUC 0.116 vs. 0.049), and the curve plateaus near eight features.

5.7. Attribution Stability as a Drift Signal

Fig. 6 tracks global importance across the training,

5.5. Is SHAP the Right Selection Criterion?

A natural question is whether SHAP is actually the best way to choose features, or merely one reasonable way. Table III and Fig. 5(a) compare selection criteria by their mean test PR-AUC across the feature-count grid. The honest answer is nuanced: mutual information edges out SHAP on raw predictive quality (mean PR-AUC 0.744 versus 0.710), with the standardized-coefficient criterion lower (0.677) and random subsets far below (0.463). SHAP is therefore competitive with the best filter, not strictly dominant over it. What distinguishes SHAP is that the same computation that ranks features also yields faithful per-transaction explanations and a consistent global importance—outputs a mutual-information score simply does not provide. In other words, with SGFA the explanations come for free alongside a selector as good as the strongest alternative we tested.

Table 3. Feature-selection criteria (mean test pr-auc over k)

Selector	Mean PR-AUC	Explanations?
SHAP (ours)	0.710	Local + global, faithful
Mutual information	0.744	None
Logistic coef.	0.677	Global, linear only
Random subset	0.463	None

5.6. Are the Attributions Faithful?

Fig. 5(b) reports the ablation test. Removing features in SHAP order degrades the model much faster than removing them at random: the area under the SHAP-order AOPC curve is 0.116 versus 0.049 for random order, a 2.4x ratio, confirming that the attributions are faithful to the model’s behavior. The comprehensiveness curve also flattens at around eight features, independently corroborating the $k^* = 8$ selection.

validation, and test windows. The top of the ranking is highly stable—the Spearman correlation between training and

validation importance vectors is 0.945—while agreement weakens into the more distant test window (0.733 validation-to-test), and the top-ten feature sets overlap by a Jaccard index between 0.67 and 0.82. The pattern is intuitive: the strongest

features remain strong, while lower-ranked features reshuffle as the distribution drifts. Monitoring these quantities online gives an inexpensive, model-centric early-warning signal that complements performance-based drift detectors [36], [37].

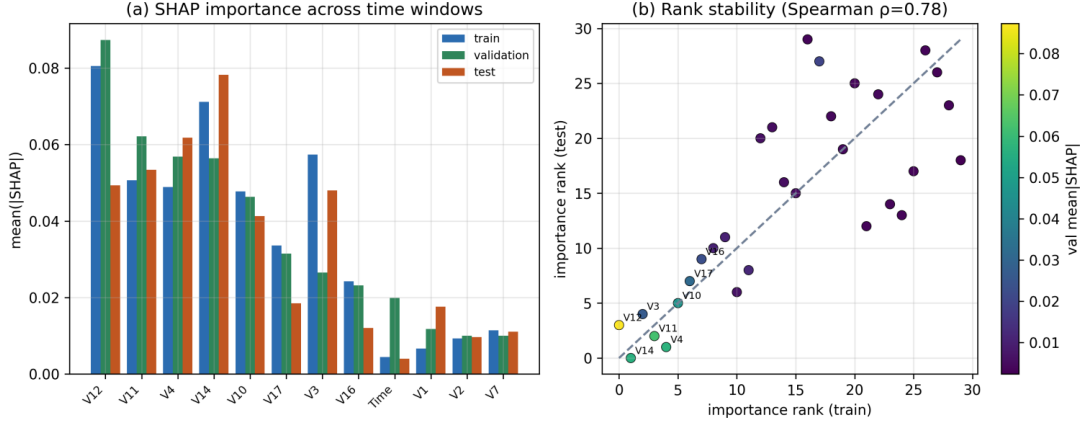


Fig. 6 Attribution stability across time windows. Left: global feature importance recomputed on the training, validation, and test windows. Right: rank agreement between windows. High-ranked features are stable while lower-ranked ones drift, so a drop in rank correlation is a practical, model-centric drift signal.

5.8. Local Explanations for Analysts

Beyond aggregate importance, SGFA supports per-decision explanations. Fig. 7 shows a SHAP waterfall for a single transaction the model flags as fraud, decomposing the score into feature contributions; a handful of features (notably V14 and V12) drive the decision. This is precisely the artifact a fraud analyst needs to triage an alert, and the kind of record a compliance process can retain to justify an automated decision [7].

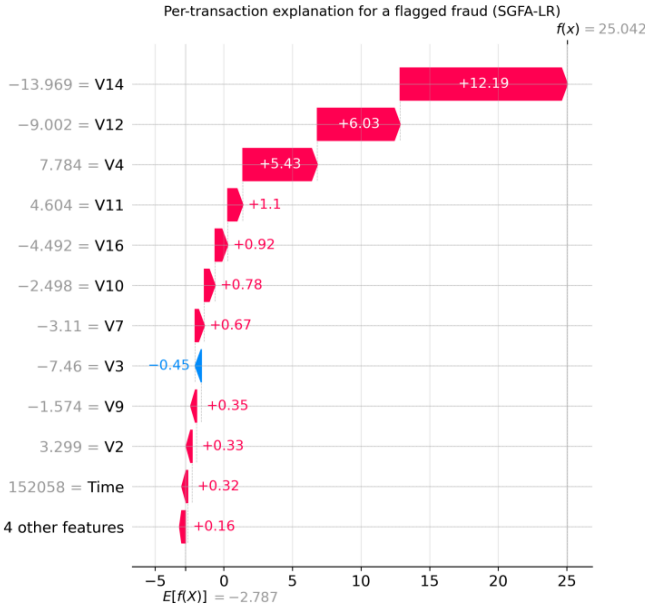


Fig. 7 Local SHAP explanation (waterfall) for a single flagged transaction, showing how each feature pushes the score above or below the base rate. Such per-decision explanations support analyst triage and regulatory justification.

6. Discussion

Our central finding is that explainability need not be a tax on a cloud-native fraud pipeline; used well, it improves the pipeline. By folding TreeSHAP attributions into feature selection, SGFA produces a model that is smaller and cheaper to serve, a feature pipeline with fewer inputs to compute and

maintain [13], and, as a byproduct, both global and local explanations suitable for analyst triage and regulatory justification [7], [6]. A logistic-regression student on eight features scores at a p99 of a few hundred microseconds, comfortably within the tail-latency budget that inline scoring demands [3].

Several honest limitations qualify these results. First, on raw predictive quality SHAP does not beat mutual information as a selector; its advantage is the joint delivery of selection and faithful explanation, not selection alone. Second, feature pruning does not shrink tree-ensemble footprint much, because that cost is structural; the footprint benefit accrues to the linear student. Third, the ULB features are PCA-anonymized, so our explanations are faithful but not semantically actionable at the level of business concepts; on raw features the same method would produce directly interpretable explanations. Fourth, we evaluate a single dataset and a single temporal split, so the drift analysis is indicative rather than conclusive. Fifth, post-hoc explanations can in principle be gamed by adversarial models [34]; our reliance on aggregate, global attributions mitigates but does not eliminate this concern. Finally, TreeSHAP on the teacher is comparatively expensive, though it runs offline.

These limitations map to future work: an online variant of SGFA that re-attributes and re-selects when stability drops; a cost-sensitive selection rule that weighs each feature’s computation cost against its attribution; deployment on raw, semantically meaningful features; validation across multiple datasets and drift regimes; and integration with probability calibration for imbalanced data [14], [15].

7. Conclusion

We presented SHAP-Guided Feature Adaptation, a teacher-student method that uses faithful Shapley-value attributions both to compress a fraud-detection model and to monitor it. On a public credit-card benchmark, under an operating-point-aware, chronologically split protocol, SGFA reduced the feature set from thirty to eight while retaining ninety-six to ninety-eight percent of full-feature ranking quality, delivered sub-millisecond tail latency and a kilobyte-scale footprint with its linear student, matched the strongest feature selector

we tested while additionally producing faithful local and global explanations, and turned attribution stability into a practical concept-drift signal. For cloud-native payment fraud detection, where accuracy, latency, and explainability are usually seen as competing, SGFA shows that a single attribution-driven design can serve all three at once.

References

- [1] Jamshidi, P., Pahl, C., Mendonca, N. C., Lewis, J., & Tilkov, S. (2018). Microservices: The journey so far and challenges ahead. *IEEE Software*, 35(3), 24-35. <https://doi.org/10.1109/MS.2018.2141039>
- [2] Zi, B. (2024). Cloud-native distributed systems for real-time payment intelligence. *AI and Data Science Journal*, 1(1), 51-56.
- [3] Dean, J., & Barroso, L. A. (2013). The tail at scale. *Communications of the ACM*, 56(2), 74-80. <https://doi.org/10.1145/2408776.2408794>
- [4] Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2011). Data mining for credit card fraud: A comparative study. *Decision Support Systems*, 50(3), 602-613. <https://doi.org/10.1016/j.dss.2010.08.008>
- [5] Dal Pozzolo, A., Caelen, O., Le Borgne, Y.-A., Waterschoot, S., & Bontempi, G. (2014). Learned lessons in credit card fraud detection from a practitioner perspective. *Expert Systems with Applications*, 41(10), 4915-4928. <https://doi.org/10.1016/j.eswa.2014.02.026>
- [6] Goodman, B., & Flaxman, S. (2017). European Union regulations on algorithmic decision-making and a "right to explanation." *AI Magazine*, 38(3), 50-57. <https://doi.org/10.1609/aimag.v38i3.2741>
- [7] Psychoula, I., Gutmann, A., Mainali, P., Lee, S. H., Dunphy, P., & Petitcolas, F. (2021). Explainable machine learning for fraud detection. *Computer*, 54(10), 49-59. <https://doi.org/10.1109/MC.2021.3082965>
- [8] Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30 (NeurIPS)* (pp. 4765-4774).
- [9] Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., & Lee, S.-I. (2020). From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, 2(1), 56-67. <https://doi.org/10.1038/s42256-019-0138-9>
- [10] Carcillo, F., Dal Pozzolo, A., Le Borgne, Y.-A., Caelen, O., Mazzer, Y., & Bontempi, G. (2018). SCARFF: A scalable framework for streaming credit card fraud detection with Spark. *Information Fusion*, 41, 182-194. <https://doi.org/10.1016/j.inffus.2017.09.005>
- [11] Carcillo, F., Le Borgne, Y.-A., Caelen, O., & Bontempi, G. (2018). Streaming active learning strategies for real-life credit card fraud detection: Assessment and visualization. *International Journal of Data Science and Analytics*, 5(4), 285-300. <https://doi.org/10.1007/s41060-017-0076-9>
- [12] Carcillo, F., Le Borgne, Y.-A., Caelen, O., Kessaci, Y., Oble, F., & Bontempi, G. (2021). Combining unsupervised and supervised learning in credit card fraud detection. *Information Sciences*, 557, 317-331. <https://doi.org/10.1016/j.ins.2019.05.042>
- [13] Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., & Young, M. (2015). Hidden technical debt in machine learning systems. In *Advances in Neural Information Processing Systems 28 (NeurIPS)* (pp. 2503-2511).
- [14] Dal Pozzolo, A., Caelen, O., Johnson, R. A., & Bontempi, G. (2015). Calibrating probability with undersampling for unbalanced classification. In *IEEE Symposium Series on Computational Intelligence (SSCI)* (pp. 159-166). IEEE. <https://doi.org/10.1109/SSCI.2015.33>
- [15] Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., & Bontempi, G. (2018). Credit card fraud detection: A realistic modeling and a novel learning strategy. *IEEE Transactions on Neural Networks and Learning Systems*, 29(8), 3784-3797. <https://doi.org/10.1109/TNNLS.2017.2736643>
- [16] Davis, J., & Goadrich, M. (2006). The relationship between precision-recall and ROC curves. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)* (pp. 233-240). ACM. <https://doi.org/10.1145/1143844.1143874>
- [17] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, 321-357. <https://doi.org/10.1613/jair.953>
- [18] Bahnsen, A. C., Aouada, D., Stojanovic, A., & Ottersten, B. (2016). Feature engineering strategies for credit card fraud detection. *Expert Systems with Applications*, 51, 134-142. <https://doi.org/10.1016/j.eswa.2015.12.030>
- [19] Jurgovsky, J., Granitzer, M., Ziegler, K., Calabretto, S., Portier, P.-E., He-Guelton, L., & Caelen, O. (2018). Sequence classification for credit-card fraud detection. *Expert Systems with Applications*, 100, 234-245. <https://doi.org/10.1016/j.eswa.2018.01.037>
- [20] Fiore, U., De Santis, A., Perla, F., Zanetti, P., & Palmieri, F. (2019). Using generative adversarial networks for improving classification effectiveness in credit card fraud detection. *Information Sciences*, 479, 448-455. <https://doi.org/10.1016/j.ins.2018.02.030>
- [21] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5-32. <https://doi.org/10.1023/A:1010933404324>
- [22] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems 30 (NeurIPS)* (pp. 3146-3154).
- [23] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)* (pp. 785-794). ACM. <https://doi.org/10.1145/2939672.2939785>
- [24] Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In *Proceedings of the 19th International Conference on Computational Statistics (COMPSTAT)* (pp. 177-186). Springer. https://doi.org/10.1007/978-3-7908-2604-3_16
- [25] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.
- [26] Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv*, arXiv:1702.08608. <https://doi.org/10.48550/arXiv.1702.08608>
- [27] Teng, D., Rhee, M., Qin, Y., Zi, B., & Liu, W. (2026). SW-SpeedDLM: Sliding-window speculative decoding for diffusion language models under long-context constraints. *Mathematics*, 14(12), Article 2137. <https://doi.org/10.3390/math14122137>
- [28] Wang, Z., Yang, J. S., Shang, W., & Ding, J. (2026). FairPromote: Explainable and fairness-aware talent promotion prediction via adversarial debiasing and SHAP-based interpretation. *IEEE Access*, 14, 1-16.

- [29] Teng, D. (2025). TEAS: Token- and energy-aware autoscaling for cost-efficient LLM serving. *AI and Data Science Journal*, 6(3), 1-10.
- [30] Zhang, F., Guo, Z., Ding, J., Yang, J., & Liu, W. (2026). Adaptive sensor fusion for robust perception in dense fog: A gated vision and LiDAR integration framework. *Sensors*, 26(12), Article 3728. <https://doi.org/10.3390/s26123728>
- [31] Zi, B. (2024). Large language models for enterprise workflow automation in financial operations. *Innovation and Technology Studies*, 1(1), 24-29.
- [32] Ding, J., Shen, Z., & Liu, W. (2026). Game-theoretic cost-sensitive adversarial training for robust cloud intrusion detection against GAN-based evasion attacks. *Applied Sciences*, 16(8), Article 3944. <https://doi.org/10.3390/app16083944>
- [33] Wang, B., Wang, Z., Zhao, W., Zhang, F., & Shang, W. (2026). DRL-Adapt: Deep reinforcement learning for adaptive routing convergence optimization in large-scale networks. *IEEE Open Journal of the Computer Society*, 7, 1-14.
- [34] Teng, D. (2025). PACO: Predictive auto-configuration for SLO-constrained large language model inference serving. *Innovation and Technology Studies*, 2(1), 1-10.
- [35] Jiao, Y., Fan, H., Yue, X., Ping, W., Sun, T., & Wang, J. (2026). Dynamic heterogeneous graph contrastive learning for uncovering collusive financial fraud. *Scientific Reports*, 16, Article 14567. <https://doi.org/10.1038/s41598-026-58938-5>
- [36] Ping, W., Jiao, Y., Fan, H., & Zhang, X. (2026). Multimodal fraud detection in financial statements: A trimodal attention network with contrastive evidence chain construction. *IEEE Access*, 14, 1-17.
- [37] Liang, Y., Jiao, Y., Ping, W., Fan, H., & Han, X. (2026). Adaptive event-driven labeling: A neuro-symbolic multiagent framework for causal inference in non-stationary time series. *IEEE Access*, 14, 1-18.