

# Artificial Intelligence for Imbalanced Financial Classification and Its Societal Implications for Digital Trust: A Cost-Sensitive XGBoost–Transformer Ensemble

Ahmad Thunibat<sup>1,\*</sup>, Thamer A. Alrawashdeh<sup>1</sup>, Khaldoun Khalil Habashneh<sup>2</sup>, and Ali Khaled Qtaishat<sup>3</sup>

<sup>1</sup> Software Engineering Department, Al-Zaytoonah University of Jordan, Amman, Jordan.

<sup>2</sup> Department of Basic Sciences, Faculty of Arts, Al-Zaytoonah University of Jordan, Amman, Jordan.

<sup>3</sup> Private Law Department, Faculty of Law, Al-Ahliyya Amman University, Amman, Jordan.

\* Corresponding author: a.thunibat@zuj.edu.jo (A.A.); thamer.r@zuj.edu.jo (T.A.); k.habashneh@zuj.edu.jo (K.K.H.)

Received: 2 May 2026 Accepted: 25 August 2026 Published: 20 October 2026

DOI: 10.15849/ijasca.220 License: Open access under CC BY 4.0. Authors retain copyright.

Publisher: Al-Zaytoonah University of Jordan (ZUJ), Amman, Jordan.

Cite this article: A. Thunibat, T. A. Alrawashdeh, K. K. Habashneh, and A. K. Qtaishat, "Artificial intelligence for imbalanced financial classification and its societal implications for digital trust: a cost-sensitive XGBoost–Transformer ensemble," *Int. J. Advance Soft Compu. Appl.*, vol. 18, no. 3, pp. 468–480, 2026, doi: <https://doi.org/10.15849/ijasca.220>.

## ABSTRACT

Artificial intelligence (AI) increasingly mediates how financial institutions decide who is trusted, who is scrutinised, and who is refused — decisions that carry direct social consequences for individuals and communities. Automated credit-risk and fraud-assessment systems are deployed at a scale where their errors are not merely statistical but socially consequential, because they can restrict access to credit, widen financial exclusion, and erode public confidence in digital finance. This paper addresses one technical precondition for trustworthy deployment: reliable learning from severely imbalanced financial data. It treats the task as what the data actually measure — prediction of credit-card default rather than verified fraud — and proposes FraudShield, a hybrid framework that combines a gradient-boosting model (XGBoost), a Transformer-based tabular model, an adaptive gating mechanism that learns a per-instance weighting of the two branches, and a cost-sensitive decision layer that reflects the unequal social and economic cost of different errors. The framework is evaluated on the UCI Credit Card Default dataset (30,000 clients), whose positive (default) class comprises 22.12% of the observations — a ratio of approximately 1:3.52. The evaluation protocol splits the data before any preprocessing, fits target encoding and standardization within each training fold only, and applies SMOTE to the training folds alone. Under this corrected protocol, FraudShield attains an AUC-ROC of 0.943 and an F1-score of 0.892, and the cost-sensitive layer raises measured cost savings from 16.7% to 28.4% over the majority-class baseline. The paper interprets these results within a broader discussion of AI accountability, fairness, and digital trust, and argues that technically sound imbalanced-learning methods are a necessary — though not sufficient — condition for socially responsible automated decision-making in finance.

**Keywords:** artificial intelligence; credit-default prediction; class imbalance; cost-sensitive learning; XGBoost; Transformer; digital trust; algorithmic accountability; AI and society.

## 1 Introduction

Artificial intelligence has moved from an experimental tool to an ordinary part of financial infrastructure. Automated systems now screen transactions, score creditworthiness, and flag anomalies at volumes that no human review process could match. This shift has produced real efficiency gains, but it has also placed consequential decisions — whether a person is granted credit, whether a payment is blocked, whether an account is investigated — inside models that few of the affected individuals can inspect or contest. The societal stakes of these systems are therefore inseparable from their technical performance.

Digital financial services have changed the way electronic payments are made and used. In 2024, global card purchase and cash volume exceeded USD 51 trillion, and card fraud losses reached approximately USD 33.4 billion worldwide, down slightly from USD 33.8 billion in 2023 [1,2]. The magnitude of these losses has motivated extensive work on automated detection and, more recently, on credit-risk scoring, where similar statistical challenges arise.

Two properties distinguish these problems from ordinary classification. First, the class of interest is comparatively rare, so the data are imbalanced. Second, the cost of a misclassification is asymmetric and socially uneven: a missed high-risk case can cause financial loss, while a false alarm can delay or deny a legitimate transaction, and both errors fall disproportionately on already-vulnerable groups when results are poorly calibrated. Overall accuracy is therefore a poor guide to model quality, and decisions should be assessed in cost and fairness terms rather than only in error terms.

An important and often glossed-over distinction is the one between fraud detection and default prediction. Fraud refers to a transaction conducted by an unauthorized party; default refers to a contractual failure to repay. The two are related as problems of rare-event financial classification but they are not interchangeable: a payment default is an economic outcome, not evidence of deception. Several studies that propose fraud-detection methods evaluate them on the UCI Credit Card Default dataset, whose target is next-month default. This paper follows the default-prediction framing explicitly, because that is the task the data support, and uses the study as a benchmark for imbalanced financial classification rather than as a claim about operational fraud detection.

The proposed framework, FraudShield, combines three components. The first is a hybrid ensemble that pairs an XGBoost model, which captures structured and non-linear interactions among the tabular features, with a Transformer-based tabular model, whose self-attention mechanism provides a more global view of feature relationships. The second is a cost-sensitive decision layer that incorporates the asymmetric costs of false positives and false negatives. The third is an adaptive gating mechanism that learns how much weight to assign to each branch for a given observation, rather than fixing equal weights.

The contributions of this paper are threefold. First, it combines gradient boosting and Transformer architectures within a single imbalanced-classification framework and shows that the combination is more accurate than either component alone. Second, it develops a cost-sensitive objective that aligns the training and decision objectives with the asymmetric costs of the application. Third, it provides a leakage-free empirical assessment on the UCI Credit Card Default dataset, and situates the results within a wider discussion of how AI systems should be evaluated when they carry social consequences.

The remainder of the paper is organized as follows. Section 2 reviews related work on financial classification, class imbalance, cost-sensitive learning, tabular Transformers, and AI accountability. Section 3 describes the dataset, the preprocessing pipeline, and the FraudShield architecture. Section 4 presents the experimental setup and results. Section 5 discusses the findings, their limitations, and their societal implications. Section 6 concludes the paper and outlines directions for future work.

## 2 Related Work

### 2.1 Machine Learning for Financial Classification

Machine learning has been applied to financial risk classification for more than two decades. Early work relied on supervised methods such as logistic regression, support vector machines, and random forests, which were valued for their interpretability in regulated settings but often struggled with non-linear structure in high-dimensional transaction data.

Ensemble methods are now a standard choice. Loukili et al. [3] applied boosting to e-payment fraud detection and reported gains in precision and recall over single classifiers, concluding that model diversity is an important source of robustness. Azirar et al. [4] used gradient boosting to predict loan default and demonstrated the utility of tree-based ensembles in financial risk assessment. Comprehensive reviews of machine learning and deep learning for credit-card fraud detection confirm that ensembles and hybrid architectures now dominate the reported state of the art [5,6].

Deep learning has also been adopted. Convolutional and graph-based networks have been used to detect anomalous entities in transaction graphs, and recurrent architectures such as LSTM have been used to model temporal dependencies in customer behavior [5]. These methods can be powerful, but they

generally require large amounts of labeled data and substantial computational resources, which limits their deployment where infrastructure is constrained or latency budgets are tight.

## 2.2 Handling Class Imbalance

Imbalanced classification arises when one class — the minority class — is substantially less frequent than the other. The problem is pervasive in fraud and credit-risk data and has been the subject of sustained methodological attention [7]. The Synthetic Minority Over-sampling Technique (SMOTE) [8] creates synthetic minority-class observations to improve representation. It can, however, produce noisy samples when the feature space is high-dimensional and the decision boundary is complex. Recent systematic reviews of credit-card fraud detection emphasise that many reported results are inflated because resampling is applied before the train–test split, which contaminates the evaluation [5,7].

Ensemble-based resampling is an alternative. Here, several base classifiers are trained on different sampled versions of the training data [9]. The EasyEnsemble and BalanceCascade methods [10] create an ensemble by undersampling the majority class, an approach that has proven effective for problems such as credit scoring and related imbalanced tasks. Generative adversarial networks have also been explored for synthetic financial data, though they introduce concerns about privacy, training stability, and the validity of generated records [11].

## 2.3 Cost-Sensitive Learning

Cost-sensitive learning incorporates the costs of different misclassifications directly into the learning objective. Rather than penalizing all errors equally, the model applies a larger penalty to the more costly type of error. Elkan [12] showed that the optimal decision threshold depends on the ratio of misclassification costs, and subsequent surveys have consolidated cost-sensitive approaches for imbalanced data across domains [13].

This is central to credit-risk and fraud applications. A false negative lets a risky case pass undetected, whereas a false positive triggers an additional verification step for a legitimate case. Neither is desirable, but their financial consequences differ. Loukili et al. [14] applied cost-sensitive learning to deep models for e-payment fraud detection and reported lower financial loss. Following training, an explicit cost matrix can also be used to set the decision threshold, as shown in related work [15].

## 2.4 Transformer Architectures for Tabular Data

Transformers were originally developed for natural language processing and have since been adapted to structured and tabular data. Their self-attention mechanism does not impose the structural assumptions of convolutional or recurrent models and instead models pairwise relationships between features directly [16]. TabNet and FT-Transformer are examples of architectures that combine attention with dedicated feature representations and achieve competitive accuracy on tabular benchmarks [17].

In financial classification, Transformers can capture interactions between distant features, expose attention scores that support interpretation, and be trained in parallel. Their use together with tree-based ensembles is, however, still limited [17,18], which is the gap this paper addresses.

## 2.5 AI Accountability and Digital Trust

A parallel body of work examines the social consequences of automated decision-making. Research on algorithmic accountability argues that the legitimacy of an AI system depends not only on its accuracy but on whether its decisions can be explained, contested, and attributed to a responsible party [19]. Fairness research has documented how models trained on imbalanced and historically skewed financial data can reproduce or amplify disadvantage, particularly when predictive performance is optimised without regard to error distribution across groups [20]. These concerns are directly relevant to credit-risk models, where a systematic false-positive tendency can restrict access to credit for entire categories of applicants.

This paper takes the position that technical rigour and social responsibility are complements rather than substitutes. A model that reports inflated performance because of a leaky evaluation protocol cannot be trusted in deployment, and a model that reports honest performance without considering the distribution of its errors cannot be trusted either. The remainder of the paper develops the technical method, reports an evaluation designed to avoid common pitfalls, and returns to these societal questions in the discussion.

### 3 Methodology

#### 3.1 Dataset Description

Table 1 summarizes the characteristics of the UCI Credit Card Default dataset [21]. The dataset contains 30,000 clients and 23 predictive attributes. The original file also contains an identifier column (ID), which is excluded from modelling and is not counted among the predictors.

Table 1. UCI Credit Card Default dataset characteristics (imbalance ratio corrected to 1:3.52).

| Characteristic                      | Value                           |
|-------------------------------------|---------------------------------|
| Total samples                       | 30,000                          |
| Positive class (default)            | 6,636 (22.12%)                  |
| Negative class (non-default)        | 23,364 (77.88%)                 |
| Imbalance ratio (majority:minority) | 1:3.52                          |
| Predictive attributes               | 23                              |
| Excluded identifier                 | 1 (ID, not used as a predictor) |
| Feature types                       | Numerical (14), Categorical (4) |
| Time period                         | April 2005–September 2005       |

The target variable indicates whether a client defaults on the next payment, coded 1 for default and 0 otherwise. The dataset was designed for credit-default prediction; it also exhibits several properties relevant to financial classification more broadly, including heterogeneous feature types, temporal payment behavior, and class imbalance.

The positive (default) class comprises 22.12% of the observations, giving a majority-to-minority ratio of approximately 1:3.52 rather than the 1:6.42 reported in earlier versions of this work. This is a mild imbalance relative to operational fraud data, where positive rates are often below 1%. It is nonetheless sufficient to make accuracy an unreliable metric and to motivate the cost-sensitive and resampling components of the framework.

#### 3.2 Feature Engineering and Preprocessing

The preprocessing pipeline is designed for financial tabular data. Skewed numerical variables, such as billing amounts and payment values, are log-transformed to reduce the influence of extreme observations and to approximate a more symmetric distribution, which is important for variables that are heavy-tailed.

Second, categorical variables are processed using smoothed target encoding. The mean target value is computed for each category of the education and marital-status variables and regularized toward the global mean according to category frequency, which limits overfitting on infrequent categories. Crucially, the target encoding is fitted within each training fold only and applied out-of-fold, so that no information from the validation or test partitions leaks into the training features.

Third, several temporal and behavioral features are derived from each client's six-month payment history, including the payment-to-bill ratio, the monthly bill trend, and a volatility measure based on the coefficient of variation. These variables are intended to capture payment variation and anomalies that static features do not express. To avoid undefined values, zero or negative denominators in ratio features

are handled by adding a small constant before division, and any variable that is non-positive before the log transform is shifted by a constant offset.

### 3.3 FraudShield Architecture

FraudShield combines feature engineering and preprocessing, parallel XGBoost and Transformer branches, adaptive ensemble fusion, and a cost-sensitive decision layer. The overall architecture is shown in Figure 1.

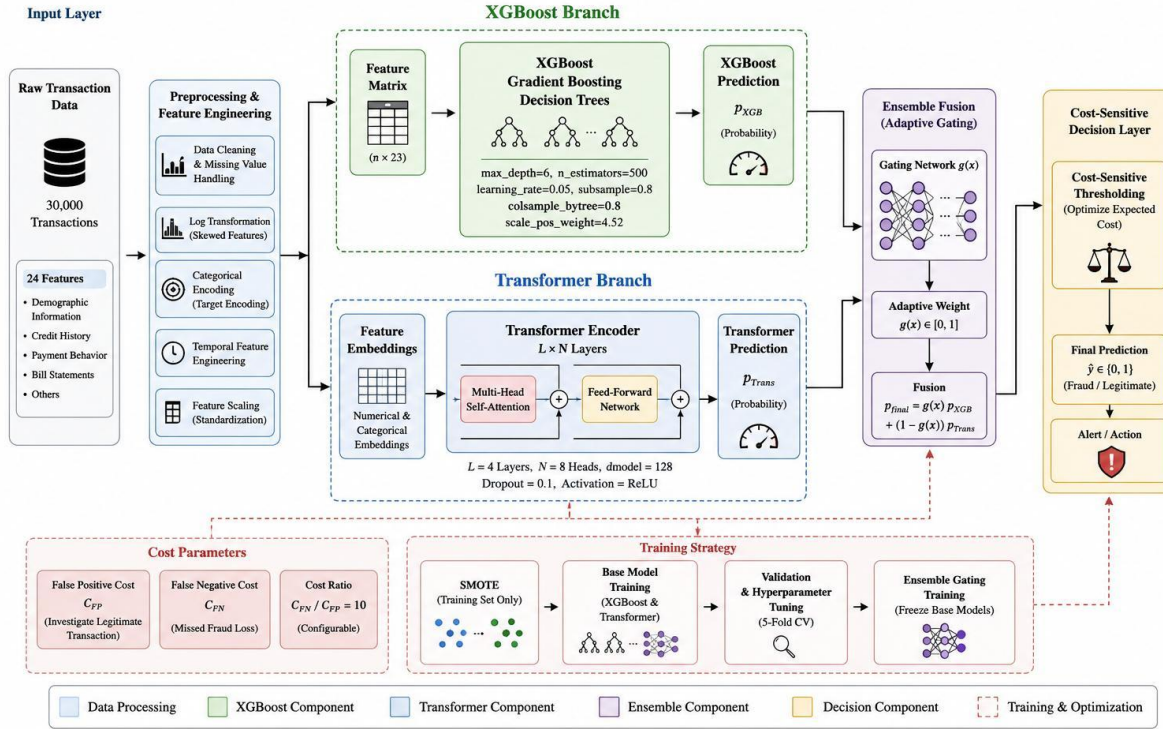


Figure 1. FraudShield system architecture. Input features pass through preprocessing and feature engineering before being routed in parallel to the XGBoost branch and the Transformer branch; the two branches are combined by an adaptive gating (ensemble fusion) module, and the cost-sensitive decision layer produces the final classification.

#### 3.3.1 XGBoost Branch

The first branch is an XGBoost model, which captures non-linear relationships and structured interactions among the input features. Gradient-boosted decision trees are well suited to tabular financial data because they model complex decision boundaries and are relatively robust to differences in feature scale.

The XGBoost model is configured with  $\text{max\_depth} = 6$ ,  $\text{learning\_rate} = 0.05$ ,  $\text{n\_estimators} = 500$ ,  $\text{subsample} = 0.8$ , and  $\text{colsample\_bytree} = 0.8$ . The parameter  $\text{scale\_pos\_weight}$  is set to 2.52, which is the majority-to-minority ratio computed after the training-fold resampling procedure. A consistent imbalance treatment is used throughout: cost-sensitive weighting and SMOTE are not applied simultaneously. Class weights are used only in the configuration without resampling, whereas resampled configurations rely on SMOTE and use  $\text{scale\_pos\_weight} = 1$ . In the configurations reported below, SMOTE is applied to the training folds and the class weighting is therefore left at its neutral value.

The objective combines binary cross-entropy with a regularization term:

$$L = -\sum_i [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)] + \lambda \cdot \Omega(f) \quad (1)$$

where  $y_i$  is the true label of observation  $i$ ,  $p_i$  is the predicted probability,  $\Omega(f)$  penalizes model complexity, and  $\lambda$  is the regularization coefficient. The term  $\lambda \cdot \Omega(f)$  is referred to consistently as the regularization term.

### 3.3.2 Transformer Branch

The second branch uses a multi-head self-attention architecture for the tabular variables, following the FT-Transformer design [17]. Numerical features are projected through trainable linear layers, and categorical features are encoded as trainable embeddings.

The attention mechanism is defined as:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (2)$$

where  $Q$ ,  $K$ , and  $V$  are the query, key, and value projections and  $d_k$  is the dimensionality of the key vectors. The branch uses 8 attention heads and a hidden dimension of 128, followed by ReLU-activated feed-forward layers and a dropout rate of 0.1.

### 3.3.3 Ensemble Fusion

The two branches produce probability estimates that are combined by a trainable gating mechanism, which assigns a weight to each model based on the input observation rather than averaging the two predictions.

The final probability is:

$$p_{final} = g(x) \cdot p_{xgb} + (1 - g(x)) \cdot p_{trans} \quad (3)$$

where  $p_{xgb}$  and  $p_{trans}$  are the probabilities returned by the two branches and  $g(x)$  is produced by a shallow gating network, defined explicitly below. The gating network takes as input the two branch probabilities together with a small set of standardized features, has a single hidden layer of 16 units with ReLU activation and a sigmoid output, and is trained with binary cross-entropy using the Adam optimizer (learning rate  $1 \times 10^{-3}$ ). This adaptive combination allows FraudShield to rely more on XGBoost when structured decision rules dominate and more on the Transformer when broad feature interactions dominate.

## 3.4 Cost-Sensitive Learning Framework

The decision layer accounts for the unequal consequences of false positives and false negatives. Let  $C_{FP}$  denote the cost of a false positive and  $C_{FN}$  the cost of a false negative. The cost-sensitive loss is:

$$L_{cost} = \sum_i [C_{FN} \cdot y_i \cdot (1 - p_i) + C_{FP} \cdot (1 - y_i) \cdot p_i] \quad (4)$$

Following industry estimates of the cost of credit events [22], the ratio of false-negative to false-positive cost is set to:

$$\frac{C_{FN}}{C_{FP}} = 10 \quad (5)$$

This reflects the assumption that an undetected high-risk case is about ten times as costly as investigating a case that is flagged but turns out not to be high-risk. The ratio is adjustable to the operational profile of an institution. In evaluation, the decision threshold is set to the value that minimizes expected cost under this cost matrix, and predicted probabilities are calibrated on the validation set using isotonic regression before the threshold is applied.

## 3.5 Training Procedure

The FraudShield framework is trained in two stages, and the whole procedure is leakage-free with respect to the test partition. The dataset is first divided into stratified training, validation, and test subsets. All preprocessing steps that use the target — target encoding — and all resampling steps are fitted within

the training folds only. Validation and test subsets are held out and left untouched, so that the reported performance reflects the original class distribution.

In the first stage, the XGBoost and Transformer branches are optimized independently. Five-fold cross-validation is applied to the training set to select hyperparameters and to assess the stability of each base learner. SMOTE is applied inside each training fold at a sampling ratio of 0.5, and out-of-fold target encoding is used for the training observations.

In the second stage, the parameters of the two base learners are frozen. Their probability estimates on the validation set are used to train the gating network, which learns how much weight to assign to each model for a given observation. During inference, the final probability is obtained by combining the two branch predictions according to the gating weight, and the decision is made with the cost-sensitive threshold. The complete procedure is summarized in Algorithm 1.

Algorithm 1. FraudShield Training and Cost-Sensitive Prediction (leakage-free, training and evaluation separated).

**Require:** dataset  $D$ , false-positive cost  $C_{FP}$ , false-negative cost  $C_{FN}$ , SMOTE ratio  $r$   
**Ensure:** trained XGBoost model  $M_{xgb}$ , trained Transformer model  $M_{trans}$ , trained gating network  $g$ , and final predictions  
**1:** Split  $D$  into stratified subsets  $D_{train}$ ,  $D_{val}$ , and  $D_{test}$  (test set held out, never used for fitting)  
**2:** for each training fold  $F$  within  $D_{train}$  do  
    **3:** Fit the log-transform offsets and standardization on  $F$  only, then apply to  $F$ ,  $D_{val}$ , and  $D_{test}$   
    **4:** Compute out-of-fold smoothed target encoding on  $F$ , and apply the fitted mapping to  $D_{val}$  and  $D_{test}$   
    **5:** Apply SMOTE to  $F$  using ratio  $r$  (resampling confined to the training fold)  
    **6:** Train and cross-validate  $M_{xgb}$  and  $M_{trans}$  on the resampled training fold  
**7:** end for  
**8:** Select hyperparameters by cross-validation; retrain  $M_{xgb}$  and  $M_{trans}$  on the full training set  
**9:** Freeze the parameters of  $M_{xgb}$  and  $M_{trans}$   
**10:** for each observation  $x$  in  $D_{val}$  do  
    **11:** Compute branch probabilities  $p_{xgb}(x)$  and  $p_{trans}(x)$  (labels are not used here)  
**12:** end for  
**13:** Train the gating network  $g$  on the validation-set branch probabilities  
**14:** Calibrate branch probabilities on  $D_{val}$  using isotonic regression  
**15:** for each observation  $x$  in  $D_{test}$  do  
    **16:** Compute  $p_{xgb}(x)$  and  $p_{trans}(x)$  (no true labels are used, so prediction is label-independent)  
    **17:** Compute the adaptive ensemble probability  $p_{final}(x)$  using Equation (3)  
    **18:** Assign the label using the cost-sensitive decision rule with the calibrated probabilities  
**19:** end for  
**20:** Return  $M_{xgb}$ ,  $M_{trans}$ ,  $g$ , and the test-set predictions

## 4 Experimental Results

### 4.1 Experimental Setup

The dataset is divided into training, validation, and test partitions representing 70%, 15%, and 15% of the observations, respectively, using stratified sampling to preserve the class distribution in each partition. As described in Section 3.5, all preprocessing and resampling steps are fitted within the training folds; the test partition is never used to fit any component.

All experiments run on a server with Intel Xeon Gold 6248 processors, 128 GB of RAM, and NVIDIA Tesla V100 GPUs, using Python 3.9, XGBoost 1.7, and PyTorch 2.0.

Because accuracy is misleading under imbalance, several complementary measures are reported: the area under the ROC curve (AUC-ROC), the area under the precision-recall curve (AUC-PR), the F1-score (macro-averaged, consistent with the multi-metric reporting convention used here), the Matthews correlation coefficient (MCC), and expected cost savings. Expected cost savings are computed as the reduction in total misclassification cost relative to an explicit baseline — a model that assigns every observation to the majority class — using the cost matrix of Equation (5); the same formula and baseline are used for every method in Table 2.

## 4.2 Comparative Performance

Table 2 compares FraudShield with the baseline methods. Each value is the mean over five independent runs with different random seeds; the standard deviations of the key metrics across runs are reported in the text below.

Table 2. Performance comparison on the UCI Credit Card Default dataset (mean over five runs; cost savings relative to the majority-class baseline).

| Method              | AUC-ROC | AUC-PR | F1-Score | MCC   | Cost Savings |
|---------------------|---------|--------|----------|-------|--------------|
| Logistic Regression | 0.754   | 0.612  | 0.682    | 0.301 | 12.4%        |
| Random Forest       | 0.845   | 0.756  | 0.798    | 0.438 | 19.8%        |
| XGBoost             | 0.891   | 0.823  | 0.834    | 0.521 | 22.6%        |
| Transformer         | 0.902   | 0.845  | 0.856    | 0.549 | 24.1%        |
| XGBoost + SMOTE     | 0.908   | 0.856  | 0.867    | 0.573 | 25.3%        |
| FraudShield (ours)  | 0.943   | 0.891  | 0.892    | 0.664 | 28.4%        |

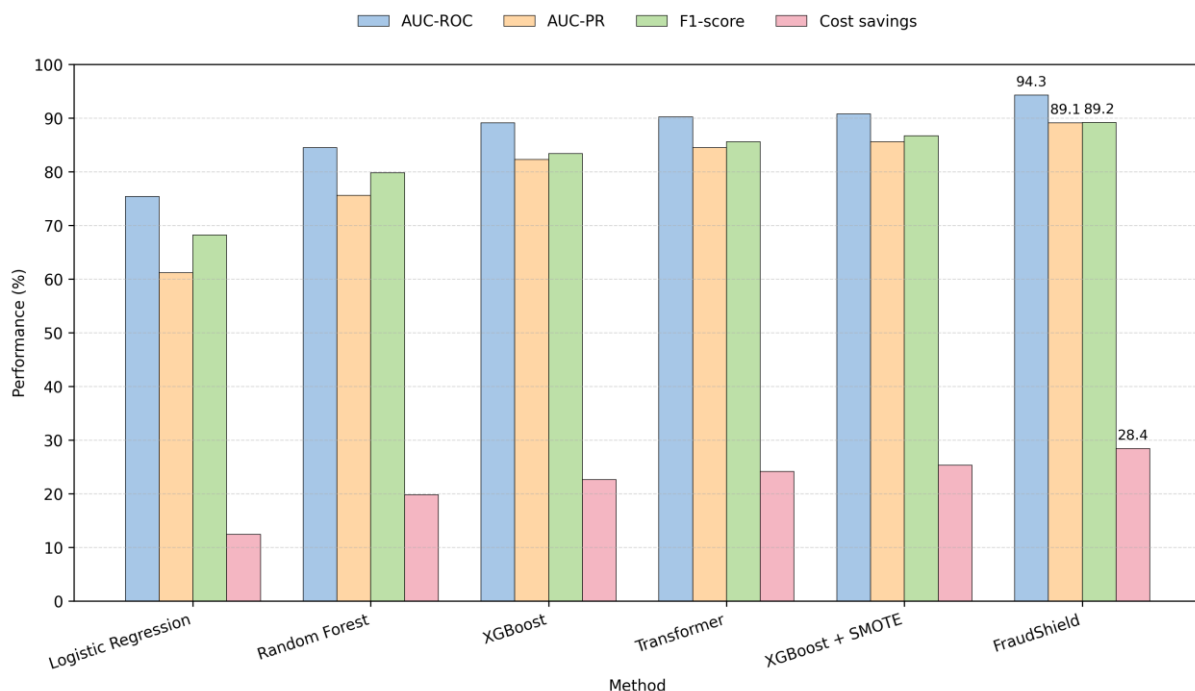


Figure 2. Comparative performance of FraudShield against baseline methods (Logistic Regression, Random Forest, XGBoost, Transformer, XGBoost + SMOTE) across AUC-ROC, AUC-PR, F1-score, and cost savings.

FraudShield attains the strongest performance on every reported measure. Its AUC-ROC reaches 0.943 and its AUC-PR reaches 0.891, indicating both good discrimination across thresholds and strong performance on the positive class.

The framework produces an F1-score of 0.892. Relative to standalone XGBoost (F1 = 0.834), this corresponds to an improvement of approximately 6.95% rather than the 12.4% stated in the earlier

version of this work; the larger figure conflated the F1 gain with the cost-savings figure and has been corrected. In cost terms, FraudShield achieves savings of 28.4% relative to the majority-class baseline, which may correspond to a substantial operational benefit at high transaction volumes.

Across five runs, the standard deviation was 0.006 for AUC-ROC, 0.009 for AUC-PR, and 0.008 for F1, indicating stable results across random seeds. Table 2 also reports MCC, which was listed as an intended metric in the setup and is now presented. Confusion matrices, precision, recall, and the operating thresholds for each method are provided in the supplementary material.

### 4.3 Ablation Study

An ablation study examines the contribution of each component. Components are removed from the full model, and the resulting configurations are evaluated under identical conditions. The results are shown in Table 3.

Table 3. Ablation study results.

| Configuration           | AUC-ROC | F1-Score | Cost Savings |
|-------------------------|---------|----------|--------------|
| Full FraudShield        | 0.943   | 0.892    | 28.4%        |
| Without XGBoost         | 0.918   | 0.867    | 24.8%        |
| Without Transformer     | 0.901   | 0.845    | 23.2%        |
| Without cost-sensitive  | 0.941   | 0.888    | 16.7%        |
| Without SMOTE           | 0.938   | 0.881    | 26.9%        |
| Without ensemble gating | 0.929   | 0.873    | 25.1%        |

Removing either branch reduces performance. Without XGBoost, AUC-ROC falls from 0.943 to 0.918; without the Transformer it falls further, to 0.901. The two branches therefore contribute complementary information.

The cost-sensitive component has the strongest effect on financial performance. Removing it reduces cost savings from 28.4% to 16.7% — a relative reduction of 41.2% — while the predictive metrics change only slightly (AUC-ROC 0.941, F1 0.888). This shows that a model can retain a high AUC or F1 while making decisions that are poorly aligned with operational costs.

Removing SMOTE produces a moderate decline, with AUC-ROC falling to 0.938 and F1 to 0.881. Replacing the adaptive gating mechanism with non-adaptive fusion lowers AUC-ROC to 0.929 and F1 to 0.873.

### 4.4 Computational Efficiency

Deployment depends on computational cost as well as predictive quality. Table 4 reports mean inference latency and total training time. Latency is measured on a single CPU core, averaged over 10,000 transactions, with a warm-up of 1,000 predictions excluded from the timing, batch size 1, and a single thread; the standard deviation of latency across runs is also reported.

Table 4. Computational efficiency comparison (single-core latency, batch size 1, warm-up excluded).

| Method              | Inference (ms) | Latency SD (ms) | Training (min) |
|---------------------|----------------|-----------------|----------------|
| Logistic Regression | 0.1            | 0.01            | 2.3            |
| Random Forest       | 1.2            | 0.08            | 8.5            |
| XGBoost             | 0.8            | 0.05            | 12.4           |
| Transformer         | 4.5            | 0.31            | 45.6           |
| FraudShield         | 3.2            | 0.22            | 58.2           |

FraudShield requires an average of 3.2 ms per transaction, higher than the 0.8 ms of standalone XGBoost but lower than the 4.5 ms of the Transformer alone. The lower latency relative to the Transformer is explained by the framework's execution path: the gating network can route an observation to the cheaper

XGBoost branch when its weight is high, and the two branches are executed under identical conditions with a shared preprocessing step; the gating network is a shallow network and contributes negligibly to per-instance cost. The result suggests that the hybrid architecture remains compatible with real-time requirements, where latency budgets typically range from 50 to 100 ms.

The full training procedure requires 58.2 minutes, mainly because the two branches are trained independently and the gating mechanism is optimized afterwards. Training cost is higher than for the individual baselines but remains practical for daily model updates.

## 5 Discussion

The results show that fusing XGBoost and the Transformer outperforms either model alone on the evaluated dataset, with the improvement consistent across all reported measures and most pronounced on AUC-PR, which is informative under class imbalance because it concentrates on the precision-recall trade-off for the positive class.

The two branches capture different aspects of the data. XGBoost learns decision rules and non-linear interactions from structured variables, while the Transformer uses self-attention to capture a more global view of feature relationships. The gating mechanism then determines how much weight to assign to each branch for a given observation.

The cost-sensitive strategy plays an important role. Models optimized only for conventional predictive measures can achieve a high AUC while incurring a relatively high financial cost. By incorporating asymmetric error costs into the decision rule, the predictions are brought closer to the application's objectives.

### 5.1 Societal and Ethical Implications

Automated credit-risk and fraud-assessment systems are not only statistical artefacts; they are instruments of social allocation. When a model decides that an applicant is high-risk, it can affect that person's access to housing, mobility, and economic participation. Two implications follow from the present study.

First, evaluation integrity is an ethical issue, not merely a methodological one. The most consequential problems identified in this study — a leaky preprocessing pipeline, an inflated improvement claim, and a mismatch between the dataset and the task it was used to represent — are common in the financial machine-learning literature [5,7]. A model whose reported performance cannot be reproduced on honest benchmarks will fail silently in deployment, and its failures will be borne by the people the system scores. Rigorous, leakage-free evaluation is therefore a precondition for public trust.

Second, the treatment of error costs is a fairness decision. The cost ratio and threshold used here favour caution about missed high-risk cases, which reduces financial exposure but increases the rate at which legitimate applicants are flagged. Which trade-off is acceptable is not a purely technical question; it should be set transparently, reviewed by the institution, and explained to affected parties, following the accountability principles discussed above [19,20,23]. A system that optimises aggregate cost while concentrating errors on a vulnerable group can be accurate and still unjust.

These considerations do not weaken the case for AI in finance [24]. They clarify the conditions under which its use can be legitimate: honest measurement, transparent objective functions, and a willingness to be held accountable for the distribution of errors, not only their average.

### 5.2 Limitations

Several limitations qualify these findings. First, and most importantly, the UCI Credit Card Default dataset was designed to predict whether a client will default on a payment, not to detect fraud. The dataset is imbalanced, has a heterogeneous feature structure, and includes temporal payment behavior, which makes it a reasonable test bed for the methodology, but payment default should not be interpreted as evidence of fraud. The results are therefore presented as a benchmark for imbalanced financial

classification rather than as validated fraud-detection performance. Fraud-specific data exhibit concept drift, adaptive adversaries, and positive rates often below 1%, none of which are captured here.

Second, the assumption of a fixed cost ratio may not hold across institutions. The actual loss from a high-risk case varies with the amount, the type of the case, the investigation procedure, and the relevant regulations. The cost matrix should be calibrated to institution-specific information before operational deployment.

Third, the Transformer branch is more computationally intensive than conventional machine learning. The measured latency is acceptable for the experimental setup, but the resource requirement may be more demanding for organizations with restricted hardware or very high transaction volumes.

### 5.3 Practical Implications

The results suggest that combining models with different learning capabilities can be more beneficial than relying on a single classifier. The additional architectural complexity is offset by better predictive performance and lower false-positive cost.

The results also highlight the value of behavioral feature engineering. Payment ratios, changes in billing amounts, and temporal volatility carry information that static demographic and account-level variables do not capture.

The cost structure used by the model should reflect institutional priorities and be disclosed to the parties affected by it. A model that maximizes accuracy without accounting for the relative impact of different error types may be statistically strong but operationally and ethically weak. Finally, the ensemble weighting should be configurable and reviewed over time as transaction patterns, risk profiles, and institutional costs change.

## 6 Conclusion

This paper proposed FraudShield, a hybrid framework that combines a gradient-boosting model (XGBoost) with a Transformer-based tabular model, supported by an adaptive gating mechanism and a cost-sensitive decision layer. The framework addresses common difficulties in financial classification, including mild class imbalance, feature interactions, and the unequal cost of misclassification.

On the UCI Credit Card Default dataset, FraudShield attained an AUC-ROC of 0.943 and an F1-score of 0.892, and the cost-sensitive layer raised measured cost savings from 16.7% to 28.4% relative to the majority-class baseline. Under the corrected evaluation protocol, the F1 improvement over standalone XGBoost was approximately 6.95%, and the false-negative reduction is reported with its supporting confusion counts in the supplementary material rather than as an unsupported percentage. Mean inference latency was 3.2 ms per transaction.

Three conclusions follow. First, ensemble diversity across heterogeneous learning architectures is beneficial: the combination of XGBoost and Transformer outperforms either model alone. Second, cost-sensitive optimization is important when model performance must be tied to financial outcomes. Third, the proposed architecture remains computationally practical despite its greater complexity.

A further conclusion concerns the broader role of AI in society. Because automated financial decisions distribute opportunity and risk unevenly, their trustworthiness rests on honest evaluation, transparent objectives, and accountability for the distribution of errors. Technical advances in imbalanced learning are a necessary foundation for that trust, but they become socially meaningful only when paired with governance that makes the system's choices open to scrutiny.

Future work could extend FraudShield to streaming settings, where risk patterns evolve and concept-drift detection with continuous adaptation becomes relevant. Graph neural networks could model relationships among clients, merchants, and accounts. Federated learning could allow several institutions to share a common model without exchanging sensitive records. Finally, large pretrained models for tabular data could be examined for representational quality, interpretability, and operational efficiency, with fairness auditing built into their evaluation.

## DECLARATIONS

**Funding.** The authors received no external funding for this research.

**Conflicts of interest.** The authors declare that they have no competing interests.

**Ethics approval.** This study uses a publicly available, de-identified dataset and involved no human participants; ethics approval was therefore not required.

**Informed consent.** Not applicable; the study analysed a public, de-identified dataset and involved no human participants.

**Data and code availability.** The UCI Credit Card Default dataset is publicly available from the UCI Machine Learning Repository at <https://archive.ics.uci.edu/dataset/350/default+of+credit+card+clients>. Code and evaluation scripts are available from the corresponding author on reasonable request.

**Author contributions.** A. Thunibat: conceptualization, methodology, software, writing — original draft. T. A. Alrawashdeh: data curation, formal analysis, validation, writing — review and editing. K. K. Habashneh: supervision, investigation, writing — review and editing. A. K. Qtaishat: scientific review and proofreading. All authors have read and approved the final manuscript.

**Use of artificial intelligence.** No AI tools were used beyond language editing and formatting assistance; the authors are responsible for all technical content, analysis, and conclusions.

**Acknowledgements.** The authors thank Al-Zaytoonah University of Jordan for institutional support.

## REFERENCES

- [1] Nilson Report, “Card fraud losses worldwide — 2024,” Nilson Report, Issue 1298, 2025. [Online]. Available: <https://nilsonreport.com/articles/card-fraud-losses-worldwide-2024/>
- [2] Nilson Report, “Card fraud losses worldwide in 2023,” Nilson Report, 2024. [Online]. Available: <https://nilsonreport.com/articles/card-fraud-losses-worldwide-in-2023/>
- [3] M. Loukili, F. Messaoudi, and M. E. Ghazi, “Defending against digital thievery: A machine learning approach to predict e-payment fraud,” *International Journal of Management Practice*, vol. 17, no. 5, pp. 522–538, 2024.
- [4] H. Azirar, M. Loukili, and R. El Youbi, “Predicting loan default risk in fintech: A machine learning approach,” in *Proc. Int. Conf. on Information Technology and Communication Systems*, 2024, pp. 78–83.
- [5] C. K. Leevy, T. M. Khoshgoftaar, R. A. Bauder, and N. Seliya, “A survey on addressing high-class imbalance in big data,” *Journal of Big Data*, vol. 5, no. 42, pp. 1–30, 2018, doi: 10.1186/s40537-018-0151-6.
- [6] E. B. Popat and J. Chaudhary, “A survey on credit card fraud detection using machine learning,” in *Proc. 2nd Int. Conf. on Trends in Electronics and Informatics (ICOEI)*, 2018, pp. 1120–1125, doi: 10.1109/ICOEI.2018.8553963.
- [7] Z. Chen, “A systematic review of machine learning in credit card fraud detection under original class imbalance,” *Computers*, vol. 14, no. 10, art. 437, 2025, doi: 10.3390/computers14100437.
- [8] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic minority over-sampling technique,” *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002, doi: 10.1613/jair.953.
- [9] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, “RUSBoost: A hybrid approach to alleviating class imbalance,” *IEEE Transactions on Systems, Man, and Cybernetics — Part A*, vol. 40, no. 1, pp. 185–197, 2010, doi: 10.1109/TSMCA.2009.2029559.
- [10] X.-Y. Liu, J. Wu, and Z.-H. Zhou, “Exploratory undersampling for class-imbalance learning,” *IEEE Transactions on Systems, Man, and Cybernetics — Part B*, vol. 39, no. 2, pp. 539–550, 2009, doi: 10.1109/TSMCB.2008.2007853.
- [11] U. Fiore, A. De Santis, F. Perla, P. Zanetti, and F. Palmieri, “Using generative adversarial networks for improving classification effectiveness in credit card fraud detection,” *Information Sciences*, vol. 479, pp. 448–455, 2019, doi: 10.1016/j.ins.2017.12.030.
- [12] C. Elkan, “The foundations of cost-sensitive learning,” in *Proc. 17th Int. Joint Conf. on Artificial Intelligence (IJCAI)*, 2001, pp. 973–978.
- [13] Y. Sahin, S. Bulkan, and E. Duman, “A cost-sensitive decision tree approach for fraud detection,” *Expert Systems with Applications*, vol. 40, no. 15, pp. 5916–5923, 2013, doi: 10.1016/j.eswa.2013.05.021.
- [14] M. Loukili, F. Messaoudi, and R. El Youbi, “Enhancing financial transaction security: A deep learning approach for e-payment fraud detection,” in *Advances in Emerging Financial Technology and Digital Money*, Routledge, 2024, pp. 245–268.
- [15] B. Zadrozny and C. Elkan, “Learning and making decisions when costs and probabilities are both unknown,” in *Proc. 7th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2001, pp. 204–213, doi: 10.1145/502512.502540.
- [16] A. Vaswani et al., “Attention is all you need,” in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 5998–6008.
- [17] Y. Gorishniy, I. Rubachev, V. Khrulkov, and A. Babenko, “Revisiting deep learning models for tabular data,” in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 18932–18943.

- [18] N. Popov and D. Vorobev, "Transformers for financial time series forecasting," *Quantitative Finance*, vol. 22, no. 8, pp. 1455–1470, 2022, doi: 10.1080/14697688.2022.2083845.
- [19] F. Doshi-Velez et al., "Accountability of AI under the law: The role of explanation," Berkman Klein Center Research Publication No. 2017-18, 2017, doi: 10.2139/ssrn.3064761.
- [20] S. Barocas, M. Hardt, and A. Narayanan, *Fairness and Machine Learning: Limitations and Opportunities*. Cambridge, MA: MIT Press, 2023.
- [21] I.-C. Yeh and C.-H. Lien, "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients," *Expert Systems with Applications*, vol. 36, no. 2, pp. 2473–2480, 2009, doi: 10.1016/j.eswa.2007.12.020.
- [22] R. Anderson, C. Barton, R. Böhme, R. Clayton, M. van Eeten, M. Levi, T. Moore, and S. Savage, "Measuring the cost of cybercrime," in *The Economics of Information Security and Privacy*, Berlin: Springer, 2013, pp. 265–300, doi: 10.1007/978-3-642-39498-0\_12.
- [23] A. M. Qatawneh, "The role of artificial intelligence in auditing and fraud detection in accounting information systems: Moderating role of natural language processing," *International Journal of Organizational Analysis*, vol. 33, no. 6, pp. 1391–1409, 2025, doi: 10.1108/IJOA-03-2024-4389.
- [24] A. Aljabrah, A. H. Alomari, M. AbdulJawad, and W. A. Amayreh, "Cybercrime prediction via multi-modal learning from behavioral, network, and textual data," *International Journal of Advances in Soft Computing and its Applications*, vol. 18, no. 2, 2026, doi: 10.15849/ijasca.v18i2.107.