Detecting Credit Card Fraud: A Machine Learning Approach

Sajal Das Shovon*
Dept. of Computer Science
University of Memphis
sshovon@memphis.edu

Md Al Imran H Sharker*
Dept. of Mechanical Eng.
University of Memphis
mhshrker@memphis.edu

Sunil Tamang*
Dept. of Mechanical Eng.
University of Memphis
stamang1@memphis.edu

***Abstract*—Credit card fraud detection is a rare-event classification task with extreme class imbalance and asymmetric error costs. This paper presents an imbalance-aware machine learning pipeline using a public anonymized transaction dataset. We perform exploratory analysis, compare multiple supervised models (logistic regression variants, decision tree, random forest, XGBoost, and a class-weighted multilayer perceptron), and evaluate using metrics appropriate for imbalanced data: precision, recall, F1, Matthews correlation coefficient (MCC), ROC-AUC, average precision (AP), and Brier score. Tree-based methods achieve the strongest overall balance at the default threshold, while threshold tuning using an F_2 objective (favoring recall) identifies an improved operating point for deployment. Finally, feature importance and SHAP explanations support interpretability for the best-performing models. The presented workflow provides practical guidance for building accurate and controllable fraud detection systems under severe imbalance.**

***Index Terms*—fraud detection, imbalanced classification, SMOTE, XGBoost, precision–recall, threshold tuning, SHAP**

I. Introduction

Credit card fraud is a persistent and costly problem for banks and payment networks, intensified by the rapid growth of online and card-not-present transactions. While the goal is simple—flag fraudulent activity before it causes loss—the learning task is challenging because real transaction data is extremely imbalanced. Fraud cases typically represent less than 0.2% of all transactions, meaning a model can appear "accurate" by predicting almost everything as legitimate while still missing most fraud. As a result, traditional machine learning models trained without special care often become biased toward the majority class and perform poorly at detecting rare fraudulent events.

A practical fraud detection system must also manage an important trade-off between different types of errors. Missing a fraudulent transaction (false negative) can directly lead to financial loss, while incorrectly flagging a legitimate transaction (false positive) can inconvenience customers, increase operational workload, and reduce trust in the payment system. Therefore, the objective is not just high overall accuracy, but a balanced and controllable performance that achieves strong fraud recall while keeping false alarms at an acceptable level.

*All authors contributed equally to this work.

This project addresses both the technical and practical aspects of fraud detection. We build an end-to-end machine learning pipeline that prevents data leakage, handles class imbalance through cost-sensitive learning and resampling strategies, and compares multiple model families—from linear classifiers to ensemble methods and neural networks. We evaluate models using metrics designed for imbalanced classification (such as precision, recall, F1-score, and precision–recall curves) and tune decision thresholds to select an operating point aligned with real-world needs. Finally, we include interpretability analysis so that model decisions can be explained, validated, and trusted in a financial risk setting.

II. Related Work

Credit-card fraud detection is a classic rare-event problem: genuine transactions dominate while fraud represents a tiny minority, making plain accuracy misleading and causing many standard classifiers to bias toward the majority class. Foundational work in imbalanced learning summarizes two major responses—data-level rebalancing (over/under-sampling) and algorithm-level changes (cost-sensitive learning, ensembles)—and emphasizes using evaluation measures beyond accuracy for skewed datasets [7]. A widely used rebalancing technique is SMOTE, which synthesizes minority-class samples to improve learning of the fraud class without simply duplicating rare examples [4]. For assessment, precision–recall (PR) analysis is often more informative than ROC curves under severe imbalance, because it explicitly captures the trade-off between fraud capture (recall) and false alarms (precision) [6]. In addition, probability outputs from many models can be miscalibrated; calibration methods and proper scoring rules such as the Brier score help quantify and improve probabilistic reliability, which is important for decision thresholding in practice [10].

On the modeling side, modern fraud studies frequently rely on tree ensembles and gradient boosting due to strong performance on tabular transaction data [8]. In particular, XGBoost is widely used because it scales well and supports class-weighting for imbalance [5]. Many applied papers on credit-card datasets report that Random Forests and boosted trees outperform linear baselines, especially when combined with resampling (e.g., SMOTE) [12, 11], while other works explore streaming or windowed retraining to reflect evolving transac-

tion patterns [3]. Deep learning has also been investigated, with mixed conclusions: neural models can be competitive but often increase training cost and reduce transparency compared with tree-based methods [1]. Because financial decisions require trust and auditability, interpretability has become a major topic; SHAP provides a unified feature-attribution approach for explaining individual predictions from complex models like boosted trees [9], and recent surveys highlight explainability, privacy, and operational constraints as central concerns in real deployments [2] .

A recurring research gap across many applied studies is that models are often compared using a limited metric set (commonly accuracy/ROC-AUC), and explicit threshold optimization—which are crucial for rare-event detection—are not jointly analyzed [7, 4, 10]. Moreover, although resampling is frequently used, fewer works provide a clean, side-by-side study of class weighting vs SMOTE vs under-sampling across multiple model families under a unified pipeline. Addressing these gaps motivates a framework that evaluates diverse models with imbalance-aware training, PR-focused evaluation, calibrated probabilities, and interpretable explanations to support reliable fraud screening .

III. Dataset and Exploratory Data Analysis

A. Dataset

The dataset contains numerical predictors `V1--V28` (anonymized), a relative timestamp `Time` in seconds, and transaction `Amount`. The binary label `Class` indicates legitimate (0) or fraud (1). Missing labels are dropped before modeling .

B. Class Imbalance and Train/Test Split

Fraud is extremely rare : approximately 0.1727% of transactions are labeled fraud. We use a stratified train/test split (80/20) to preserve class proportions. Standardization is fit on the training split only and applied to the test split to prevent leakage .

C. Correlation and Low-Dimensional Views

A sampled correlation heatmap suggests limited strong pairwise correlations. PCA and t-SNE projections show that fraud examples are sparse and not cleanly separable in low dimensions, motivating non-linear supervised models.

D. Temporal Fraud Analysis

The `Time` attribute is converted from seconds to hours:

$$h_i = \frac{\text{Time}_i}{3600},$$

and reduced modulo 24 to obtain hour-of-day patterns. We compute fraud histograms, fraud rates in fixed-width bins,

$$r_b = \frac{N_b^{(1)}}{N_b^{(0)} + N_b^{(1)}},$$

and compare Day 1 and Day 2 fraud percentages. KDE-based density estimates are generated for legitimate and fraudulent time distributions, enabling comparison of temporal fraud concentration.

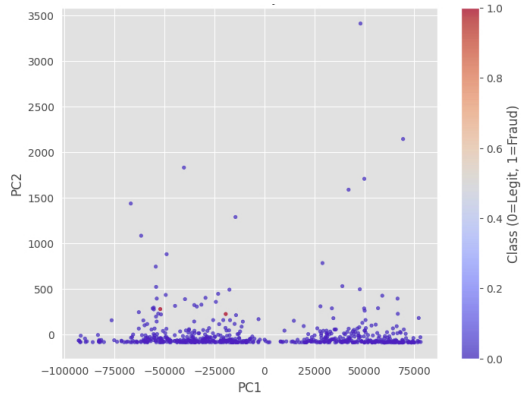

Fig. 1: Two-dimensional PCA projection of transaction features (PC1 vs. PC2).

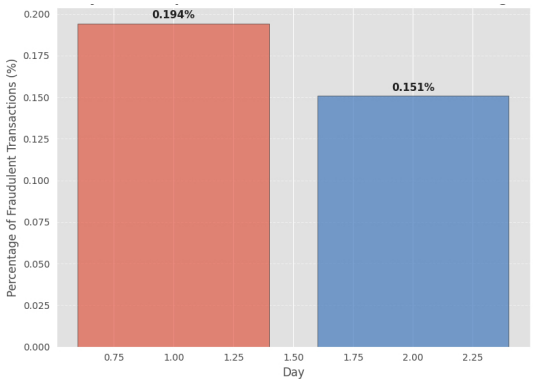

Fig. 2: Day-wise comparison of fraudulent transaction percentage

IV. Methodology

We have implemented a full supervised learning pipeline for extreme class-imbalance classification, consisting of structured preprocessing, temporal and manifold analysis, imbalance mitigation, model training, threshold optimization, and interpretability. All experiments are conducted in Python using `scikit-learn`, `imbalanced-learn`, `xgboost`, and `TensorFlow/Keras`.

A. Data Preprocessing

The dataset is imported from a CSV file into a `pandas` DataFrame. The target variable is the binary attribute `Class` ($y \in \{0, 1\}$), where 1 denotes fraud. All remaining columns, including PCA-derived components $V_1, \ldots, V_{28}$, `Time`, and `Amount`, form the predictor set X. Rows with missing labels are removed.

B. Feature Scaling

Since many models in this study are gradient-based or distance-sensitive, we have applied feature standardization. The `StandardScaler` transforms each feature x_j as

$$\tilde{x}_j = \frac{x_j - \mu_j}{\sigma_j},$$

where μ_j and σ_j are computed only from X_{train}. The fitted scaler is applied to X_{test}. Standardization is required for Logistic Regression, MLP optimization, PCA, and t-SNE.

C. Manifold Structure Analysis: PCA and t-SNE

To assess geometric separability, we have sampled up to 6000 instances and perform:

PCA.: We have computed the top two principal components by eigen-decomposition of the covariance matrix:

$$\Sigma = \frac{1}{n-1}(\tilde{X} - \bar{X})^\top (\tilde{X} - \bar{X}),$$

and project each sample using

$$Z_{\text{PCA}} = \tilde{X} W_{(:,1:2)}.$$

The resulting 2D embedding indicates global variance structure and verifies that fraud points do not form linearly separable clusters.

t-SNE.: We have employed `TSNE` with PCA initialization and adaptive learning rate to obtain a nonlinear embedding preserving local neighborhoods. This highlights minority clusters that PCA cannot detect. Both visualizations confirm the absence of simple decision boundaries.

D. Imbalance Mitigation Strategies

Given the extreme imbalance ($\approx 0.17\%$ fraud), we have applied multiple algorithmic strategies:

1) Class Weighting: For Logistic Regression and MLP, class weights are computed as:

$$w_0 = \frac{N}{2N_0}, \quad w_1 = \frac{N}{2N_1},$$

where N_0, N_1 are class counts in y_{train}. These weights rescale the loss function,

$$\mathcal{L} = -\sum_{i=1}^{N} w_{y_i} \left[y_i \log \hat{p}_i + (1 - y_i) \log(1 - \hat{p}_i) \right],$$

penalizing misclassification of minority samples.

2) Random Undersampling: We have randomly sampled the majority class to match the minority class cardinality,

$$|\mathcal{X}_{\text{maj}}^{\text{undersampled}}| = |\mathcal{X}_{\text{min}}|,$$

and train Logistic Regression on the balanced subset.

3) SMOTE Oversampling: Using the `SMOTE` algorithm, synthetic minority samples are generated as

$$x_{\text{new}} = x_i + \lambda(x_i^{(k)} - x_i), \quad \lambda \sim U(0,1),$$

where x_i is a minority point and $x_i^{(k)}$ is one of its k nearest minority neighbors. This produces a smoother minority manifold.

4) XGBoost Cost-Sensitive Reweighting: For XGBoost, we have set

$$\texttt{scale\_pos\_weight} = \frac{N_0}{N_1},$$

modifying gradient updates so that minority samples contribute proportionally more.

E. Model Architectures

We have evaluated a diverse set of models:

Logistic Regression: Implemented with `max_iter=1000`. Three variants are trained: unweighted, class-weighted, and undersampled.

Decision Tree: A depth-constrained tree with Gini impurity.

Random Forest: An ensemble of 300 trees with bootstrap sampling. Both unweighted and class-weighted versions are trained.

Bagging Ensemble: A 100-estimator `BaggingClassifier` with decision trees regularized by `max_depth=6`, `min_samples_split=20`, and `min_samples_leaf=10`.

XGBoost: A gradient-boosting model with:

$$\texttt{max\_depth} = 6, \quad \texttt{n\_estimators} = 400,$$

and cost-sensitive weighting as described above.

Multi-Layer Perceptron: A Keras network:

$$128 \xrightarrow{\text{ReLU}} \text{Dropout}(0.3) \to 64 \xrightarrow{\text{ReLU}} \text{Dropout}(0.2) \to 1 \text{ (sigmoid)},$$

trained with class-weighted binary cross-entropy, batch size 2048, 12 epochs, and validation split 0.2. AUC is tracked as a training metric.

F. Evaluation Metrics

For each model we compute:

$$\text{Precision} = \frac{TP}{TP + FP},$$

$$\text{Recall} = \frac{TP}{TP + FN},$$

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

$$\text{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}},$$

as well as ROC-AUC, confusion matrices, ROC curves, and Precision–Recall curves.

V. Results and Discussion

Table I summarizes the performance of all evaluated models at the default decision threshold $\tau = 0.5$. Metrics include Precision, Recall, F1-score, Matthews Correlation Coefficient (MCC), ROC–AUC, Brier score, and training time. These jointly characterize discriminative performance, calibration, and computational cost.

TABLE I: Performance of All Models on the Test Set

Model	Prec.	Rec.	F1	MCC	AUC	Brier
RF	0.9718	0.7340	0.8364	0.8444	0.9324	0.000525
Decision Tree	0.9718	0.7340	0.8364	0.8444	0.8511	0.000536
XGBoost	0.9114	0.7660	0.8324	0.8352	0.9704	0.000504
RF (balanced)	0.9710	0.7128	0.8221	0.8317	0.9275	0.000529
LR (plain)	0.8125	0.5532	0.6582	0.6699	0.9516	0.000928
MLP	0.0926	0.8511	0.1670	0.2778	0.9434	0.012369

A. Performance Summary

The ensemble-based classifiers exhibit the strongest performance. Random Forest (RF) achieves the highest overall effectiveness with an F1-score of 0.8364, an MCC of 0.8444, and excellent calibration (Brier = 0.000525). Although the Decision Tree attains the same Precision and Recall, its lower ROC–AUC indicates weaker ranking capability and higher variance. XGBoost delivers the best ROC–AUC (0.9704), the strongest minority-class Recall (0.7660), and competitive F1 and MCC, reflecting its ability to model complex nonlinear patterns.

Balanced RF slightly reduces Recall relative to standard RF, indicating that class weights increase model conservativeness. Among linear models, plain Logistic Regression (LR) provides moderate Precision and Recall, but its imbalance-handled variants (balanced LR, SMOTE-LR, and undersampled LR) overcorrect for class imbalance, yielding very high Recall (≈ 0.85) but severely degraded Precision (< 0.08), resulting in poor F1 and MCC.

The multilayer perceptron (MLP) also exhibits extreme Recall inflation (0.8511) with very low Precision (0.0926), a common failure mode in neural networks trained with strong class weights; although its ROC–AUC remains high (0.9434), the decision boundary becomes overly biased toward predicting the minority class, leading to elevated false positives.

B. Confusion Matrix Summary Across All Models

Table II provides a consolidated comparison of the confusion matrix entries—true positives (TP), false negatives (FN), false positives (FP), and true negatives (TN)—for all evaluated models. This tabular summary highlights how different modeling strategies trade off between detecting minority-class fraud cases and avoiding false alarms.

TABLE II: Confusion Matrix Summary for All Models

Model	TP	FN	FP	TN
LR (plain)	52	42	42	50,961
Decision Tree	69	25	0	51,003
Random Forest	69	25	0	51,003
RF (balanced)	67	27	0	51,003
XGBoost	72	22	21	50,982

Summary Interpretation.: Ensemble-based models (Decision Tree, Random Forest, XGBoost) achieve the strongest operational performance, combining high true-positive rates with minimal false positives. In contrast, linear models with imbalance-handling techniques (balanced LR, LR) dramatically inflate false-positive counts despite improving recall, often making them unsuitable for deployment where false alarms carry operational cost. Plain Logistic Regression is conservative, yielding low false positives but missing many fraud cases. XGBoost provides the best balance among all models, capturing more fraud cases while keeping false alarms low.

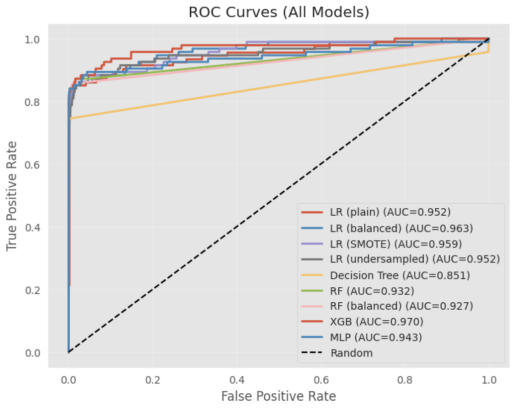

Fig. 3: ROC Curves for All Models. XGBoost achieves the highest AUC

C. ROC–AUC Analysis

The ROC curves in Fig. 3 compare the true positive rate (TPR) and false positive rate (FPR) across all trained models. Several observations can be made:

- **XGBoost achieves the best discriminative performance (AUC = 0.970).** Its ROC curve consistently dominates the others, confirming its ability to learn expressive nonlinear boundaries that separate fraudulent and legitimate transactions effectively.
- **Logistic Regression variants exhibit similar ROC shapes (AUC = 0.952–0.963).** Although sampling-based variants (balanced, SMOTE, undersampling) differ significantly in precision and recall at fixed thresholds, their ROC curves remain similar because ROC–AUC is threshold-independent. The balanced LR model shows the highest AUC among the LR methods (0.963).
- **Tree-based models show moderate AUC values.** Random Forest (AUC = 0.932) and Balanced RF (AUC = 0.927) perform well but saturate early, showing less flexibility across thresholds compared to XGBoost. Decision Tree performs the worst among tree-based approaches (AUC = 0.851), showing sharp transitions indicative of overfitting.
- **The MLP achieves competitive performance (AUC = 0.943).** While the network captures some nonlinear structure, class imbalance and shallow architecture limit its discriminative capability relative to XGBoost.
- **High-recall, low-precision models still show strong AUC.** Models such as MLP, LR (balanced), and LR (SMOTE) achieve high AUC scores but very low precision. This occurs because ROC–AUC evaluates the *ranking* of prediction scores and is insensitive to false positives, highlighting the importance of threshold optimization for operational decisions in imbalanced datasets.
- **Low FPR across all models.** Due to extreme class imbalance (fraud rate $\approx 0.18\%$), even poorly performing models exhibit very low FPR, but their relative ROC ordering still reflects true discriminative differences.

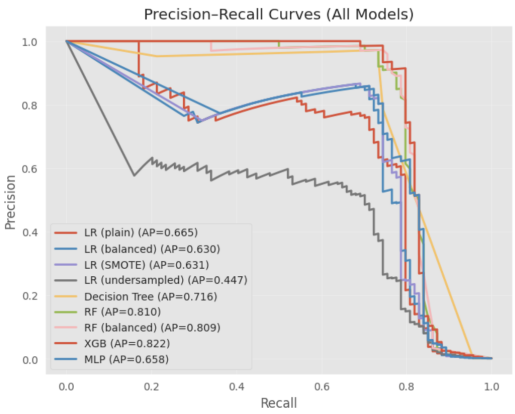

Fig. 4: Precision–Recall Curves for all models.

Overall, the ROC–AUC analysis shows that XGBoost provides the best separation between classes, followed by Logistic Regression and Random Forest models. The MLP offers moderate performance, while Decision Tree generalizes poorly. However, ROC–AUC alone is insufficient for fraud detection; models with high AUC may still exhibit poor precision, necessitating threshold tuning and cost-sensitive evaluation.

D. Precision–Recall Analysis

The Precision–Recall (PR) curves in Fig. 4 provide a more reliable assessment of model quality under extreme class imbalance than ROC curves. Because fraudulent samples constitute less than 0.2% of the dataset, PR curves emphasize the minority class performance by focusing solely on positive–class precision and recall. The following observations can be drawn:

- **XGBoost achieves the highest Average Precision (AP = 0.822).** Its PR curve remains consistently above all other models, indicating strong ranking ability and stable precision at varying recall levels. This implies that XGBoost not only separates classes well (as seen in ROC–AUC), but also maintains high precision when retrieving fraud cases.
- **Random Forest and Balanced Random Forest perform competitively (AP = 0.810 and 0.809).** These models maintain high precision for moderate recall values but drop sharply near the extreme recall region. Their strong AP values confirm robust performance under imbalance, though slightly below XGBoost.
- **Decision Tree shows moderate performance (AP = 0.716).** While its ROC–AUC was lower, the PR curve reveals that the model can still maintain reasonable precision for a subset of recall values; however, its precision deteriorates rapidly for high recall, reflecting its limited generalization.
- **Logistic Regression variants cluster together with moderate performance (AP = 0.630–0.665).** LR (plain) achieves the best AP among logistic models (0.665), despite not applying balancing techniques. Although balanced LR and SMOTE LR improve recall at fixed thresholds, their AP remains similar due to an increased number of false positives affecting precision across the threshold range.
- **Undersampled LR performs the worst (AP = 0.447).** The loss of majority-class samples severely limits decision boundary reliability, yielding unstable precision and recall. This confirms that random undersampling is detrimental for highly imbalanced, high-dimensional data.
- **MLP achieves moderate performance (AP = 0.658).** Similar to LR (plain), the MLP exhibits good recall but suffers significant precision degradation at high recall levels. This reflects that the model learns coarse nonlinear boundaries but struggles with optimizing precision without tailored regularization or deeper architectures.

Overall, the PR analysis shows that **XGBoost and Random Forest models provide the strongest fraud-detection performance**, particularly in retaining precision as recall increases. Logistic Regression models show reasonable behavior but suffer from precision collapse at high recall. The MLP captures some nonlinear patterns but remains inferior to tree-based ensembles under severe class imbalance.

E. Why Do Models Behave Differently?

Tree-based ensembles such as RF and XGBoost naturally partition feature space in ways that isolated minority-class patterns can be captured without excessively inflating false positives. XGBoost further benefits from gradient boosting and a tuned `scale_pos_weight` parameter, producing superior ranking quality. In contrast, linear models cannot represent nonlinear fraud patterns, and when reweighted through SMOTE or class-balanced loss, they shift the decision boundary aggressively toward the minority class, causing Precision collapse.

The MLP, while expressive, suffers from two factors: (1) insufficient calibration under class imbalance, and (2) lack of tabular inductive biases, making it more prone to overpredicting the rare class. These behaviors explain its extreme Recall and weak Precision despite reasonable ROC–AUC. Overall, ensemble-tree models are best suited for highly imbalanced tabular data due to their robustness, calibration, and generalization behavior.

VI. Model Interpretability via SHAP

We applied SHAP (SHapley Additive exPlanations) to the Balanced Random Forest and XGBoost models using a subset of $3{,}000$ test samples. TreeExplainer was used to obtain exact Shapley values for each ensemble model.

1) Global Feature Importance: Both models identify a consistent set of dominant predictors: **V14, V4, V12, V17, V11**, followed by **V10, V16**, and **Amount**. These variables drive the majority of the model output. In particular, V14 and V17 show the strongest contribution magnitudes, confirming that fraud samples are characterized by extreme outlier behavior in these PCA-derived components.

2) SHAP Summary Insights: The SHAP beeswarm plots reveal that:

- Large-magnitude negative values of **V14** sharply increase the fraud probability.
- High absolute deviations in **V4, V12, V17,** and **V11** consistently push predictions toward the fraud class.
- **Amount** contributes positively but less strongly, indicating that unusually large transactions moderately increase risk.

Overall, fraud is not driven by a single feature but by multi-dimensional extremal patterns across several PCA components.

3) RF vs. XGB Behavior: XGBoost exhibits more concentrated SHAP importance, forming sharper decision boundaries, consistent with its superior ROC–AUC and PR–AUC. The Balanced Random Forest distributes importance across more variables, reflecting ensemble averaging and increased robustness but slightly weaker discrimination.

4) Summary: SHAP confirms that fraud detection in this dataset relies on extreme values in multiple latent components, validating the need for nonlinear tree-based models (RF, XGB) over linear baselines.

VII. Conclusion

This work investigated credit card fraud detection under extreme class imbalance by developing a complete pipeline that integrates preprocessing, imbalance handling, exploratory analysis, and a broad suite of machine learning models. Our primary objective was to build a system capable of reliably identifying fraudulent transactions despite the rarity of the positive class, and the experimental findings confirm that this objective was achieved.

Overall, the study demonstrates that robust fraud prediction is feasible even with highly skewed data when appropriate preprocessing, resampling, and model selection strategies are employed. The combination of advanced ensemble models and interpretability techniques provides both strong predictive performance and insight into the underlying fraud patterns, fulfilling the main goal of reliable detection in an imbalanced real-world setting.

As a learning outcome, we built practical experience with the Python ML stack (NumPy/Pandas, scikit-learn, imblearn/SMOTE, XGBoost, TensorFlow/Keras) and with evaluation tools suited to imbalance (PR curves, MCC, calibration/Brier score). We also learned how decision-threshold selection affects real deployment behavior and how SHAP helps explain and justify model decisions in a high-stakes setting.

References

[1] Fawaz Khaled Alarfaj et al. "Credit card fraud detection using state-of-the-art machine learning and deep learning algorithms". In: *Ieee Access* 10 (2022), pp. 39700–39715.

[2] Rejwan Bin Sulaiman, Vitaly Schetinin, and Paul Sant. "Review of machine learning approach on credit card fraud detection". In: *Human-Centric Intelligent Systems* 2.1 (2022), pp. 55–68.

[3] Sree Charitha et al. "Credit Card Fraud Analysis Using". In: *Advances in Communication and Applications: Proceedings of ERCICA 2023, Volume 2* 1105 (2024), p. 285.

[4] Nitesh V Chawla et al. "SMOTE: synthetic minority over-sampling technique". In: *Journal of artificial intelligence research* 16 (2002), pp. 321–357.

[5] Tianqi Chen. "XGBoost: A Scalable Tree Boosting System". In: *Cornell University* (2016).

[6] Jesse Davis and Mark Goadrich. "The relationship between Precision-Recall and ROC curves". In: *Proceedings of the 23rd international conference on Machine learning*. 2006, pp. 233–240.

[7] Haibo He and Edwardo A Garcia. "Learning from imbalanced data". In: *IEEE Transactions on knowledge and data engineering* 21.9 (2009), pp. 1263–1284.

[8] Yvan Lucas et al. "Multiple perspectives HMM-based feature engineering for credit card fraud detection". In: *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*. 2019, pp. 1359–1361.

[9] Scott M Lundberg and Su-In Lee. "A unified approach to interpreting model predictions". In: *Advances in neural information processing systems* 30 (2017).

[10] Alexandru Niculescu-Mizil and Rich Caruana. "Predicting good probabilities with supervised learning". In: *Proceedings of the 22nd international conference on Machine learning*. 2005, pp. 625–632.

[11] Gajula Lakshmi Sahithi et al. "Credit card fraud detection using ensemble methods in machine learning". In: *2022 6th International conference on trends in electronics and informatics (ICOEI)*. IEEE. 2022, pp. 1237–1241.

[12] Dejan Varmedja et al. "Credit card fraud detection-machine learning methods". In: *2019 18th International Symposium Infoteh-Jahorina (Infoteh)*. IEEE. 2019, pp. 1–5.

VIII. Task Distribution

Member	Responsibilities
Sajal Das Shovon	Data preprocessing and EDA, temporal fraud analysis, PCA/t-SNE visualization, model evaluation utilities, MLP development and tuning, SHAP interpretability analysis, and final report organization.
Md Al Imran Hasan Sharker	Baseline model implementation (LR, DT), advanced models (RF, XGB), hyperparameter tuning, threshold optimization, precision–recall and ROC curve analysis, and comparative metrics evaluation across all models.
Sunil Tamang	Imbalance-handling pipeline (SMOTE, undersampling, class weights), model calibration (Brier score, reliability curves), threshold selection criteria, confusion matrix analysis for all models, supplemental experiments, and report formatting.