Credit Card Fraud Detection with Imbalanced Data Using Apache Spark

Yehong Huang*
Kennesaw State University
Kennesaw, Georgia, USA
yhuang27@kennesaw.edu

Xiaoyue Hu*
Kennesaw State University
Kennesaw, Georgia, USA
xhu5@students.kennesaw.edu

Kunal Shenoi*
Kennesaw State University
Kennesaw, Georgia, USA
kshenoi1@students.kennesaw.edu

Kokou Adje*
Kennesaw State University
Kennesaw, Georgia, USA
kadje@students.kennesaw.edu

Shivang Patel*
Kennesaw State University
Kennesaw, Georgia, USA
spate336@students.kennesaw.edu

Abstract

Credit card fraud detection is challenging because fraudulent transactions are extremely rare compared with legitimate transactions. This project studies credit card fraud detection using Apache Spark and evaluates five machine learning models: Logistic Regression, Random Forest, Gradient-Boosted Trees, Neural Network, and XGBoost. The models are compared under five imbalance-handling strategies: baseline training, class-weighted learning, random undersampling, SMOTE oversampling, and a hybrid SMOTE plus undersampling strategy. The cleaned dataset contains 283,726 transactions after duplicate removal, with only 473 fraudulent transactions, representing 0.1667% of the data. Model performance is evaluated using precision, recall, F1-score, AUC-ROC, AUC-PR, false positive rate, and the combined metric F1 × (1 - FPR). The experimental results show that baseline XGBoost achieves the strongest overall performance, while SMOTE XGBoost provides the best AUC-PR and stronger recall. These findings demonstrate that tree-based ensemble models are especially effective for imbalanced fraud detection when evaluated with metrics beyond accuracy.

1 Research Statement and Conjecture

Intelligent detection of credit card fraud is a major problem in large-scale financial systems because fraudulent transactions occur far less frequently than legitimate transactions. In real-world transaction streams, fraudulent cases typically account for less than 1% of total observations, making traditional classification methods biased toward the majority class. A classifier can appear highly accurate by predicting nearly all transactions as legitimate, while still failing to identify the rare fraud cases that matter most. Therefore, fraud detection requires models and evaluation metrics that focus directly on minority-class detection and false-positive control.

This project implements and compares multiple machine learning models using Apache Spark and related Python machine learning tools. The models include Logistic Regression, Random Forest, Gradient-Boosted Trees, Neural Network, and XGBoost. To address the imbalance problem, the study evaluates several imbalance-handling techniques, including class-weighted learning, random undersampling, SMOTE oversampling, and a hybrid balancing approach. The goal is not only to maximize accuracy, but to identify models that can detect fraud cases effectively while avoiding excessive false alarms.

We hypothesize that nonlinear ensemble-based models, especially Random Forest, Gradient-Boosted Trees, and XGBoost, will outperform a standard linear classifier trained on highly imbalanced data. These models are expected to better capture complex nonlinear fraud patterns in high-dimensional transaction features. We also expect that imbalance-handling methods will improve recall for the minority fraud class, although they may reduce precision by increasing the number of false positives. Therefore, the best model must be selected by balancing recall, precision, F1-score, AUC-PR, and false-positive control rather than accuracy alone.

2 Related Work

Credit card fraud detection has been studied extensively in the context of highly imbalanced financial data, where fraudulent transactions represent only a very small fraction of total observations. Recent work has increasingly focused on scalable machine learning pipelines that can operate efficiently on large transaction datasets while still improving sensitivity to the minority fraud class. In this area, distributed computing frameworks such as Apache Spark have become especially important because they support large-scale preprocessing, model training, and evaluation in practical time.

Alshammari et al. [1] developed a credit card fraud detection system using big data analytics and compared several machine learning models within a distributed framework. Their study showed that ensemble-based approaches produced strong classification performance, with Gradient Boosting achieving the highest overall accuracy and precision, while Random Forest provided stronger recall than several competing methods. Similarly, Jha et al. [2] evaluated Random Forest, Support Vector Machine, XGBoost, and Logistic Regression for fraud detection using Apache Spark. Their findings highlighted XGBoost as a particularly effective model because it provided a strong balance between predictive performance, efficiency, and scalability in a distributed setting.

More recent work has also explored distributed deep learning for fraud detection. Izaddoost and Chatterjee [3] presented a cloud-based distributed deep learning framework for credit card fraud detection and studied its scalability across multiple worker nodes. Their results showed that distributed training reduced runtime while maintaining strong recall, demonstrating that deep learning can be made practical for this problem when supported by scalable infrastructure. However, their work also reinforces a broader point

in the fraud detection literature: high recall alone is not sufficient if it comes at the cost of too many false positives.

Beyond batch classification studies, Carcillo et al. [4] proposed SCARFF, a scalable Spark-based framework for streaming credit card fraud detection. Their work is especially important because it moves the problem closer to real-world deployment, where fraud detection systems must process high-volume transaction streams in near real time rather than only offline historical datasets. This study shows that scalability, adaptability, and deployment context are just as important as raw predictive performance when designing fraud detection systems.

Other research has examined synthetic data generation to improve learning under severe class imbalance. Fiore et al. [5] used generative adversarial networks to improve classification effectiveness in credit card fraud detection. Their results suggest that synthetic fraud generation can help address the scarcity of minority-class examples and may improve the ability of classifiers to detect subtle fraudulent patterns. This line of work is closely related to more traditional sampling techniques such as SMOTE, since both aim to improve minority-class representation during training.

Despite these contributions, several limitations remain across the literature. Many studies still place heavy emphasis on overall accuracy, even though accuracy can be misleading in extremely imbalanced datasets where a classifier can achieve very high accuracy while missing many fraud cases. In addition, prior work often focuses on a narrow subset of models or a single imbalance-handling strategy, making it difficult to compare linear, ensemble, boosting, and neural approaches under a unified experimental framework.

Building on these studies, the present project develops a fraud detection pipeline that compares five models: Logistic Regression, Random Forest, Gradient-Boosted Trees, XGBoost, and Neural Network. In contrast to work that relies mainly on accuracy, this study emphasizes precision, recall, F1-score, AUC-ROC, AUC-PR, and the combined metric F1 × (1 - FPR). It also investigates explicit imbalance-handling strategies, including class weighting, SMOTE, random undersampling, and a hybrid resampling pipeline.

3 Methodology

This study follows a machine learning workflow implemented primarily with Apache Spark for large-scale data processing and model training. The overall methodology includes dataset loading, data cleaning, class distribution analysis, feature preparation, imbalance handling, model training, prediction, and comparative evaluation. The purpose is to measure how well different models identify the minority fraud class while limiting false positives.

3.1 Dataset and Preprocessing

This study uses the public credit card fraud dataset containing European credit card transactions. The original dataset contains 284,807 rows and 31 columns. The columns include *Time*, *Amount*, the anonymized PCA-transformed features *V1* through *V28*, and the target label *Class*. The target variable uses 0 for legitimate transactions and 1 for fraudulent transactions.

The preprocessing workflow first checks for missing values across all columns. The final run showed that all columns had zero null values. Next, duplicate records were removed. The original dataset contained 284,807 rows, with 1,081 duplicates. After removing duplicates, the cleaned dataset contained 283,726 rows. The class distribution was then profiled to quantify the imbalance. The cleaned dataset contained 283,253 legitimate transactions and 473 fraudulent transactions. This means that 99.8333% of the records were legitimate, while only 0.1667% were fraudulent.

All predictor columns were assembled into a single feature vector for model training. The cleaned data was then split into 227,171 training rows and 56,555 testing rows. The same train-test split was used across all models and balancing strategies to ensure fair comparison.

3.2 Imbalance Handling Methods

Because the fraud class was extremely rare, this project evaluated multiple imbalance-handling methods.

- **Baseline training** used the original imbalanced training data without additional balancing.
- **Class-weighted learning** assigned greater training weight to fraud cases. In the final run, the majority class weight was 0.500842 and the minority class weight was 297.344241.
- **Random undersampling** reduced the number of majority-class examples to create a more balanced training set. In the final run, the majority class was reduced from 226,789 examples to approximately 350 examples, while the minority class contained 382 examples.
- **SMOTE oversampling** generated synthetic minority-class examples. The training set was changed from 226,789 majority and 382 minority examples to 226,789 majority and 22,678 minority examples, creating an approximate 10:1 ratio.
- **Hybrid balancing** combined SMOTE oversampling with majority-class undersampling. The minority class was first increased to 22,678 examples, and the majority class was then reduced to 45,356 examples, producing an approximate 2:1 ratio.

These strategies were applied only to the training set. The test set was not resampled or reweighted, which preserves a realistic evaluation distribution and prevents data leakage.

3.3 Models

Five machine learning models were evaluated.

- **Logistic Regression** served as the linear baseline model.
- **Random Forest** represented a bagging-based ensemble tree method.
- **Gradient-Boosted Trees (GBT)** represented a boosting-based tree ensemble method.
- **Neural Network** was implemented as a multi-layer perceptron classifier.
- **XGBoost** was implemented as a gradient boosting model and was evaluated as one of the strongest nonlinear models.

Together, these models provide a comparison between linear, ensemble, boosting, and neural approaches. Some models support direct sample weighting, while others do not. For example, class weights were used where supported, but GBT and Neural Network

models did not directly use a weight column in the Spark implementation. For those models, resampling-based strategies such as SMOTE and undersampling are especially important.

3.4 Evaluation Metrics

Accuracy was not treated as the primary metric because the dataset was extremely imbalanced. A model could achieve very high accuracy by predicting nearly all transactions as legitimate. Therefore, the study used the following metrics:

- **Precision:** the proportion of predicted fraud cases that were actually fraud.
- **Recall:** the proportion of actual fraud cases correctly detected.
- **F1-score:** the harmonic mean of precision and recall.
- **Specificity:** the proportion of legitimate transactions correctly identified.
- **False Positive Rate (FPR):** the proportion of legitimate transactions incorrectly flagged as fraud.
- **AUC-ROC:** ranking performance across classification thresholds.
- **AUC-PR:** precision-recall ranking quality, especially important for imbalanced datasets.
- **F1 × (1 - FPR):** a combined project metric that rewards strong F1 performance while penalizing false positives.

The false positive rate is calculated as:

$$FPR = \frac{FP}{FP + TN}.$$

The combined metric is calculated as:

$$F1 \times (1 - FPR).$$

This metric was used to rank the final model-strategy combinations because it balances fraud detection strength with false-positive control.

4 Experimental Design

Using an experimental design, the authors compared five models with five different training strategies, resulting in 25 different model-strategy combinations. Each model was trained using the same training split and evaluated using the same held-out test split. This experimental design allowed for a direct comparison of the models as well as the methods used to handle class imbalance independently.

The authors employed several training strategies to test their models, including baseline training, class-weighted learning, random undersampling, SMOTE oversampling, and hybrid balancing. The five models that were tested included Logistic Regression, Random Forest, Gradient-Boosted Trees (GBT), Neural Network, and XGBoost. For each model-strategy combination, predictions were made on the unchanged test set, and confusion matrix values were calculated based on those predictions. From the confusion matrix values, other performance metrics were calculated, including precision, recall, F1-score, accuracy, specificity, false positive rate (FPR), F1 × (1 - FPR), area under the receiver operating characteristic curve (AUC-ROC), and area under the precision-recall curve (AUC-PR).

The authors primarily compared the results of each model using F1 × (1 - FPR). This was selected as the primary metric because fraud detection encompasses more than just maximizing recall. For example, a model that identifies many fraudulent transactions but also flags thousands of legitimate transactions as fraudulent would likely not be useful to real-world financial systems. Therefore, F1 × (1 - FPR) is a useful measure because it rewards a good balance between precision and recall while minimizing excessive false positives.

Since fraud represented only 0.1667% of the cleaned dataset, AUC-PR was also emphasized because it is particularly useful when assessing rare positive outcome classes.

5 Results

Table 1 presents the top five model and strategy combinations sorted by F1 × (1 - FPR). The best overall model was baseline XGBoost, followed closely by baseline GBT, class-weighted GBT, and baseline Random Forest.

Table 1: Top Model and Strategy Combinations

Strategy	Model	Recall	Precision	F1	F1×(1-FPR)	AUC-ROC	AUC-PR
Baseline	XGBoost	0.7363	0.9571	0.8323	0.8323	0.9657	0.7750
Baseline	GBT	0.7692	0.8974	0.8284	0.8283	0.9736	0.8101
Class Weight	GBT	0.7692	0.8974	0.8284	0.8283	0.9736	0.8101
Baseline	Random Forest	0.7473	0.9189	0.8242	0.8242	0.9836	0.7715
SMOTE	XGBoost	0.8132	0.7629	0.7872	0.7869	0.9812	0.8274

Baseline XGBoost achieved the highest combined score, with F1 × (1 - FPR) = 0.8323. It had precision of 0.9571, recall of 0.7363, F1-score of 0.8323, AUC-ROC of 0.9657, and AUC-PR of 0.7750. Its high precision shows that when it predicted fraud, it was usually correct.

Baseline GBT and class-weighted GBT both achieved F1 × (1 - FPR) = 0.8283. Their results were identical because Spark GBT did not apply the class-weight column. Both GBT configurations achieved recall of 0.7692, precision of 0.8974, F1-score of 0.8284, and AUC-PR of 0.8101.

Baseline Random Forest also performed strongly, with F1 × (1 - FPR) = 0.8242, precision of 0.9189, and recall of 0.7473. SMOTE XGBoost ranked fifth by the combined metric, but it achieved the highest AUC-PR among the top models at 0.8274 and improved recall to 0.8132.

Table 2 shows the best-performing model under each imbalance-handling strategy.

Table 2: Best Model Under Each Imbalance-Handling Strategy

Strategy	Best Model	Recall	Precision	F1	F1×(1-FPR)	AUC-PR
Baseline	XGBoost	0.7363	0.9571	0.8323	0.8323	0.7750
Class Weight	GBT	0.7692	0.8974	0.8284	0.8283	0.8101
Random Undersampling	Logistic Regression	0.8791	0.0705	0.1306	0.1282	0.5230
SMOTE	XGBoost	0.8132	0.7629	0.7872	0.7869	0.8274
Hybrid	Random Forest	0.8242	0.5769	0.6787	0.6781	0.7501

Random undersampling produced high recall but extremely low precision. The best undersampling result by the combined metric was Logistic Regression, with recall of 0.8791 but precision of only 0.0705. This indicates that the model detected many fraud cases but produced many false positives.

The hybrid strategy performed better than pure undersampling but still did not outperform the baseline or SMOTE strategies. The best hybrid model was Random Forest, with recall of 0.8242, precision of 0.5769, F1-score of 0.6787, and AUC-PR of 0.7501.

6 Analyses and Comparison

The findings indicate that accuracy alone is not sufficient for this issue. The dataset is predominantly composed of legitimate transactions, with fraudulent transactions representing only 0.1667% of the cleaned dataset. Many models appeared to be very accurate, yet there were several differences between models when comparing precision, recall, F1-score, AUC-PR, and the number of false positives.

The baseline XGBoost model produced the strongest overall results. Its final recall score was lower than some imbalance-handled models, but it had the best precision-recall balance while maintaining an extremely low false positive rate. This is particularly relevant because excessive false positives could create unnecessary friction for customers and lead to higher costs during the manual review process.

The most effective family of models in this experiment was tree-based models, including XGBoost, GBT, and Random Forest. All three types of tree-based models were among the top-performing models in the reported scores. Tree-based models likely performed well because they can model nonlinear relationships between the principal component analysis (PCA)-transformed transaction features better than a linear model such as Logistic Regression.

The use of SMOTE improved recall for multiple model types in this experiment, particularly XGBoost. The recall score for the SMOTE XGBoost model increased from 0.7363 to 0.8132 and produced the highest AUC-PR score of 0.8274. This implies that SMOTE improved the ability of the model to correctly rank transactions as fraudulent based on the model's output. However, due to using SMOTE, precision decreased from 0.9571 to 0.7629, highlighting a tradeoff between recall and precision.

Random undersampling produced very high recall values; however, performance was poor with respect to precision. For example, the undersampled Neural Network had a recall value of 0.9341, but its precision was only 0.0154. This means that while the model was able to identify many fraud cases, it incorrectly flagged a large number of legitimate transactions. As a result, random undersampling alone is not useful for deployment in this setting.

Class-weighted learning produced mixed results. This method improved recall for XGBoost compared with baseline XGBoost, but based on the F1 × (1 - FPR) metric, it did not outperform baseline XGBoost. For Logistic Regression, the model became too aggressive with its class weights, resulting in 724 false positives while lowering precision to 0.0961.

Neural Networks performed reasonably well with a recall value of 0.7253 and an F1-score of 0.7811 using the baseline settings. However, they were less competitive compared to XGBoost, GBT, and Random Forest. The Neural Network performed above average with respect to recall using random undersampling, but with such poor precision, it is not practical for deployment because it lacks reliability.

Overall, the results support the hypothesis that nonlinear ensemble methods outperform linear classification for credit card fraud detection. However, the optimal imbalance-handling strategy depends on the business objective. If the emphasis is to minimize false positives while maintaining strong fraud detection, then baseline XGBoost is the optimal choice. If the focus is to identify more fraudulent cases and the organization can tolerate additional false positives, then SMOTE XGBoost is the stronger alternative.

6.1 Summary of Key Comparisons

- The dataset was highly unbalanced, with only 0.1667% of the records being fraud cases; therefore, accuracy could not be relied on for assessing model quality.
- Baseline XGBoost was the best model in terms of the overall F1 × (1 - FPR) score, giving it the most well-rounded performance for detecting fraud and controlling false positives.
- In general, tree-based models performed better than Logistic Regression, with XGBoost, GBT, and Random Forest being among the top-performing models.
- SMOTE improved the recall and AUC-PR values for XGBoost at the cost of increased false positives, providing a clear example of the tradeoff between recall and precision.
- Random undersampling produced high recall but very low precision, making it unsuitable as a final modeling strategy.
- The use of class weights helped some models detect more fraud, but it also increased false positives, with the largest increase associated with Logistic Regression.

7 Conclusion

Apache Spark was used to detect fraud using credit card transaction data, with a focus on multiple strategies for handling imbalanced data. With only 0.1667% of records representing fraudulent transactions, the study relied mainly on precision, recall, F1-score, area under the ROC curve (AUC-ROC), area under the precision-recall curve (AUC-PR), and F1 × (1 - FPR), rather than accuracy.

The baseline XGBoost model had an F1 × (1 - FPR) score of 0.8323, with precision of 0.9571, recall of 0.7363, and an AUC-PR value of 0.7750. Thus, this model provided the best balance between detecting fraud and controlling false positive cases.

Baseline GBT, class-weighted GBT, and baseline Random Forest also had strong performance, supporting the conclusion that tree-based ensemble learners are well suited for imbalanced fraud detection problems. For the goal of maximizing recall and AUC-PR, the strongest alternative model was SMOTE XGBoost, which detected a greater number of fraud records than baseline XGBoost and had an AUC-PR value of 0.8274. However, this came with an increased number of false positive records due to decreased precision. Random undersampling was not a good method overall because it increased the number of records identified as positive but decreased precision across multiple models.

Overall, baseline XGBoost should be seen as the best model to deploy in an operational environment that balances the need for fraud detection with the minimization of false alarms. If the goal is simply to catch more fraud and the operational environment can absorb some additional false positive costs, SMOTE XGBoost could

be considered. Future work should include threshold tuning, cross-validation, hyperparameter optimization, cost-sensitive evaluation, and the use of more recent transactional data to validate model generalization.

7.1 Conclusion Bullet Points

- Baseline XGBoost was the highest-performing model based on the F1 × (1 - FPR) score.
- SMOTE XGBoost is a suitable alternative when recall and AUC-PR are more valuable than precision.
- Random undersampling alone should not be used as the sole method because it creates too many false positives.
- In this study, ensemble tree-based models outperformed both Logistic Regression and Neural Network models.
- Future improvements should include threshold tuning, cross-validation, hyperparameter optimization, and cost-sensitive evaluation.

References

[1] A. Alshammari, R. Alshammari, M. Altalak, K. Alshammari, and A. Alhakamy, "Credit-card fraud detection system using big data analytics," in *2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, 2022, pp. 1–7.

[2] P. Jha, S. Srivastava, T. Gandhi, and G. P, "Financial fraud detection for credit card transactions using apache spark," in *2024 International Conference on Sustainable Communication Networks and Application (ICSCNA)*, 2024, pp. 427–431.

[3] A. Izaddoost and A. Chatterjee, "Distributed deep learning model for credit card fraud detection: A cloud-based scalability study," in *2025 Mexican International Conference on Computer Science (ENC)*, 2025, pp. 1–6.

[4] F. Carcillo, A. D. Pozzolo, Y.-A. L. Borgne, O. Caelen, and G. Bontempi, "Scarff: A scalable framework for streaming credit card fraud detection with spark," *Information Fusion*, vol. 41, pp. 182–194, May 2018.

[5] U. Fiore, A. D. 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.