An Analysis of Machine Learning Techniques in the Field of Credit Fraud Detection in 2023

Michael Morrison
Department of Computer Science & Engineering
Texas A&M University
College Station, TX 77843 USA
mac2.morrison@tamu.edu

Aaron Weng
Department of Computer Science & Engineering
Texas A&M University
College Station, TX 77843 USA
mwkyawaw@tamu.edu

***Abstract*—In an era of rapidly evolving theft, the need for timely models has never been so apparent as it is now. If we are to prevent such unfaithful transactions, we must ensure the strategies and datasets we use are up to date. In this paper, we explore the efficacy of previously explored machine-learning techniques with a more recent dataset, and compare these strategies at a broader scale. In exploring previous papers, we found that our models not only match, but in certain cases, exceed the accuracies presented by the explored models.**

Additionally, fine-tuning these models reveals a notable improvement in their ability to generalize across datasets, highlighting their robustness in detecting unseen fraudulent patterns. The Decision Tree-based ensemble approaches (xgboost) consistently outperform other classifiers, achieving precision and recall rates exceeding 99% on datasets, however SVM achieves results even closer to 100% accuracy and near zero FPR upon hyperparameter tuning. This research underscores the importance of combining data quality improvement techniques with integrated classification frameworks to enhance the reliability and effectiveness of fraud detection systems. Furthermore, it suggests potential for broader applications of these strategies in addressing other types of anomalous behaviors, paving the way for future advancements in machine learning-driven fraud prevention.

***Index Terms*—Gradient Boosting, XGBoost, Hyperparameter Tuning, Fraud, Machine Learning**

I. Introduction

In the past few decades, credit fraud has become a significant threat in the modern financial landscape, driven by the rapid growth of digital transactions and e-commerce. Credit fraud, such as card cloning and card-not-present fraud, pose an increasingly difficult challenges to both financial institutions and consumers, and is only made worse by an increasingly digitized landscape. Despite representing a small fraction of total transactions, credit card fraud results in billions of dollars in financial losses each year and undermines trust in digital payment systems.

To address these challenges, advanced machine learning techniques have become indispensable in detecting and preventing fraudulent activities. By leveraging models like XGBoost, which excels in handling complex classification problems, and employing hyperparameter tuning to refine model performance, it is possible to enhance detection accuracy and minimize false positives. This paper explores the application of these methodologies to improve the identification of fraudulent credit card transactions, aiming to support financial institutions in staying ahead of increasingly sophisticated fraud tactics.

In this paper, we focus on a pre-created dataset from 2023 based on the transactions of european cardholders, much like the dataset used in previous studies [1]. Unlike the previous dataset though, the one discussed in this paper is much more concurrent and reflective of the current nature of credit fraud. From this dataset, we extract anonymized features based on different elements of the transaction (e.g. time, amount, location, etc.)

Initially our goal was to reproduce the results found in previous studies, while comparing different machine learning models, primarily focusing on the recent developments of XGBoost gradient boosting technology. We then fine-tuned the hyperparameters to find new ways of further improving the effectiveness of the models.

II. Related Works

The detection of credit card fraud has been extensively studied using machine learning techniques, with significant advancements in addressing challenges like class imbalance, classification boundary fuzziness, and the need for real-time processing. Previous research has explored a variety of approaches, including data augmentation, ensemble learning, and anomaly detection. The primary work our paper seeks to replicate is a study done in 2023 by M. Chen, who used comparative analysis between different machine learning models for fraud detection [1]. In his study, XGBoost consistently outperformed logistic regression, SVM, and traditional random forest models in both precision and recall metrics, demonstrating its suitability for fraud detection in complex datasets. The accuracy of all the models approach 100% accuracy, similar to ours, though the dataset used in the study was highly unbalanced and is now outdated.

Later that year, a similar study by R. Cherkaoui and E. En-Naimi explored the effectiveness of the same dataset on random forest, SVM, and K-nearest-neighbor techniques (on the supervised-learning side) [2]. Their random forest classifier reached the highest precision of the study at 95%, though the highest accuracy was shared by KNN and SVM at 95.07%. This study looked into basic hyperparameter-tuning for better results, and our work applies some of the techniques used

in this work to a different subset of models. Lastly, we can see that these two studies had a similar problem, although this was not a focus: a highly unbalanced dataset. In a study by Zhou et al., new methods for dealing with data imbalance were explored [3]. In their work they use advanced data boundary reconstruction and integrated classification to deal with their dataset, though our study circumvents these strategies entirely with a pre-cultivated balanced dataset. With supervised learning, we intended to create new models that are a better fit for real world scenarios.

III. Methodology

In this study, we aimed to evaluate the performance of three popular machine learning classifiers—Logistic Regression, Support Vector Machines (SVM), and XGBoost—for detecting fraudulent transactions. Each classifier was assessed based on its ability to distinguish between fraudulent and legitimate transactions using a dataset with labeled classes. The primary objective was to compare their classification capabilities through standard evaluation metrics and graphical analyses, including Receiver Operating Characteristic (ROC) and Precision-Recall (PR) curves.

To ensure robustness and comparability, the dataset underwent preprocessing steps such as handling missing values and scaling numerical features. The dataset was then split into training and testing subsets, with a consistent 60-40 train-test ratio maintained across all models to avoid data leakage and ensure reliable evaluation.

Each model was implemented using established libraries such as sklearn for Logistic Regression and SVM, and xgboost for gradient-boosted trees. Hyperparameters were kept at default settings for simplicity, with only necessary adjustments made (e.g., enabling probability estimation for SVM and setting evaluation metrics for XGBoost). After implementing the basic models, we performed light hyperparameter tuning on each one, singling out 1-3 parameters to find the best values for the models.

A. Dataset

The dataset used in this study contained transaction records labeled as either fraudulent or legitimate, making it a binary classification problem. It contains 568630 rows, each with 31 columns (29 features if we discount transaction id and label). The features describe the transactions, alongside a target variable (Class) that indicated whether the transaction was fraudulent (1) or not (0). Additionally, an id column uniquely identified each record. The main features of the dataset were extracted from sensitive transactional data from European credit card holders in 2023. The resulting dataset features (1 through 28) were anonymized and hidden to prevent a breach in privacy [4].

Any records with missing data were removed using the dropna() function. This ensured that only complete and reliable records were included in the analysis. The id column, which did not contribute to the classification task, was dropped. This left only the feature set (X) and target variable (y) for the analysis. To account for the varying scales of numerical features, the StandardScaler from sklearn was applied. This scaling transformed the features to have a mean of 0 and a standard deviation of 1, a necessary step for algorithms like Logistic Regression, SVM, and XGBoost to perform optimally.

Additionally, we decided to take a 10% sample of the data to help with the speed of execution. The dataset is large enough that even with only a tenth of it, there are still 56863 rows to work with. This mostly helped with the SVM runtime, but to make it consistent across all models we used this 10% sample for every one.

B. Feature Complexity

The dataset contains a variety of features, which could include numerical, categorical, and possibly missing values. These features are used to predict whether an observation belongs to a certain class (e.g., fraud or not). The complexity of the features can arise from:

- Numerical Features: These may have different scales and distributions, which can affect the performance of some machine learning models. For example, features like transaction amounts or durations may have widely varying ranges.
- Categorical Features: Some datasets might have categorical variables (e.g. transaction type) that need to be converted into numerical form using encoding methods like one-hot encoding or label encoding.
- Feature Importance: Some features might be more important than others for predicting the target variable.

C. Scaling

Scaling is an important preprocessing step for many machine learning algorithms, especially those that rely on distance metrics (e.g., Logistic Regression, SVM). For all models, the dataset was scaled using the StandardScaler:

$$x_{\text{scaled}} = \frac{x - \mu}{\sigma} \tag{1}$$

Where x_{scaled} is the scaled value of the feature, x is the original value, μ is the mean of the feature computed over all data points, and σ is the standard deviation of the feature computed over all data points. Scaling ensures that each feature contributes equally to the distance calculations in models that are sensitive to scale, preventing features with larger ranges from dominating the model's learning process.

D. Classification Models

- Logistic Regression - We first decided to go with the quintessential classifier; log regression. This is a fundamental model for binary classification, especially when the relationship between features and the target variable is assumed to be linear. It outputs probabilities (between 0 and 1) that indicate the likelihood of an observation belonging to a particular class. It has the advantages of being simple and providing probabilities for decision thresholds. In hyperparameter tuning, we adjusted the C

value, which controls the regularization strength of the model.

- Support Vector Machines (SVM) - SVM is a powerful classifier that tries to find a hyperplane that best separates the classes. It works well with both linear and non-linear data depending on different kernel functions (e.g., radial basis function or RBF kernel). SVMs are good for complex datasets and have many tuneable hyperparameters, but are computationally expensive. Because it was so computationally expensive, we opted to sample only 10% of the original data to speed up training. Through testing, we found there was very little impact on prediction accuracy from using this smaller data size. For its hyperparameters, we tested different C and gamma values. C is the regularization parameter, and gamma is a coefficient used in rbf that controls the influence of singular data points (larger means more localized, smaller means broader).
- XGBoost (Extreme Gradient Boosting) - XGBoost is an ensemble learning technique based on gradient boosting. It builds multiple weak models (decision trees) and combines them to improve accuracy. XGBoost is known for its high performance in terms of speed and accuracy, making it popular for structured/tabular data. It also handles missing data, noise, and many features well in addition to being resistant to overfitting. It is typically more computationally expensive than logistic regression. When tuning hyperparameters, we adjusted n estimators, learning rate, and max depth.
- Additional note: XGBoost was the fastest model in training and predicting by a great margin. Log regression was the second fastest, and it still took tens of times longer than XGBoost. SVM remained slower than log regression.

IV. Results (Pre-Tuning)

a) Logistic Regression: The logistic regression model was fairly accurate, with scores above 95% across the board without any hyperparameter tuning.

TABLE I
LOG REGRESSION CONFUSION MATRIX

	Predicted 0	Predicted 1
Actual 0	7841	214
Actual 1	364	7511

TABLE II
LOG REGRESSION CLASSIFICATION REPORT

Class	Precision	Recall	F1-Score	Support
0	0.96	0.97	0.96	8055
1	0.98	0.95	0.96	7875
Accuracy			0.96	15930
Macro Avg	0.96	0.96	0.96	15930
Weighted Avg	0.96	0.96	0.96	15930

b) Support Vector Machine: The SVM was very accurate, having a score of 99% across nearly every metric.

TABLE III
SVM CONFUSION MATRIX

	Predicted 0	Predicted 1
Actual 0	7970	85
Actual 1	135	7740

TABLE IV
SVM CLASSIFICATION REPORT

Class	Precision	Recall	F1-Score	Support
0.0	0.98	0.99	0.99	8055
1.0	0.99	0.98	0.99	7875
Accuracy			0.99	15930
Macro Avg	0.99	0.99	0.99	15930
Weighted Avg	0.99	0.99	0.99	15930

c) XGBoost: XGBoost was extraordinarily accurate, with a near perfect score in every manner of prediction testing, similar to previous studies [1].

TABLE V
XGBOOST CONFUSION MATRIX

	Predicted 0	Predicted 1
Actual 0	8022	33
Actual 1	6	7869

TABLE VI
XGBOOST CLASSIFICATION REPORT

Class	Precision	Recall	F1-Score	Support
0.0	1.00	1.00	1.00	8055
1.0	1.00	1.00	1.00	7875
Accuracy			1.00	15930
Macro Avg	1.00	1.00	1.00	15930
Weighted Avg	1.00	1.00	1.00	15930

Pre-tuning, XGBoost was found to be the strongest prediction model overall. This is a finding in related work, as well as log regression generally being weaker [1] [2]. The combined ROC curve as well as the precision-recall curve illustrating the difference in model results is shown below:

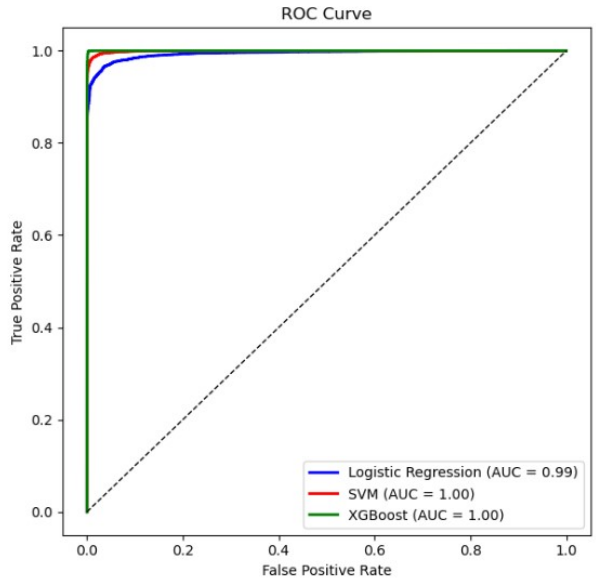

Fig. 1. The ROC Curve for all models before tuning.

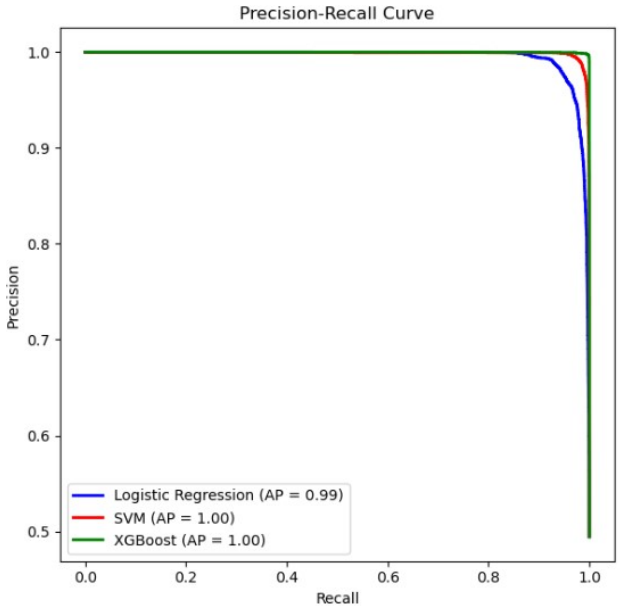

Fig. 2. The Precision-Recall Curve for all models before tuning.

V. Results (After Tuning)

a) Logistic Regression: We tried tuning the log regression model using different values of C (0.01, 0.1, 1, 10). The best result was C=10, which improved the scores greatly in some areas, though not enough to overtake the other two models.

TABLE VII
Optimized Log Regression Confusion Matrix

	Predicted 0	Predicted 1
Actual 0	7836	219
Actual 1	359	7516

TABLE VIII
Optimized Log Regression Classification Report

Class	Precision	Recall	F1-Score	Support
0.0	0.96	0.97	0.96	8055
1.0	0.97	0.95	0.96	7875
Accuracy			0.97	15930
Macro Avg	0.96	0.96	0.96	15930
Weighted Avg	0.96	0.96	0.96	15930

b) Support Vector Machine: The C (0.1, 1, 10) and gamma ('scale', 'auto', 0.1, 1') values were adjusted for SVM. It was found that C=10 and gamma=0.1 yielded the best scores. This turned the already strong SVM predictor into a near perfect one.

TABLE IX
Optimized SVM Confusion Matrix

	Predicted 0	Predicted 1
Actual 0	8021	34
Actual 1	2	7873

TABLE X
Optimized SVM Classification Report

Class	Precision	Recall	F1-Score	Support
0.0	1.00	1.00	1.00	8055
1.0	1.00	1.00	0.99	7875
Accuracy			1.00	15930
Macro Avg	1.00	1.00	1.00	15930
Weighted Avg	1.00	1.00	1.00	15930

c) XGBoost: For XGBoost, we adjusted the n estimators (50, 100, 200), learning rate (0.01, 0.1, 0.2), and max depth (3, 5, 7). We found in testing that the best combination was n_estimators=100, learning_rate=0.2, and max_depth=7. Unfortunately, our tuning resulted in a model that performed slightly worse than the original, untuned XGBoost model. Although, it still performed incredibly well on the test dataset.

TABLE XI
Optimized XGBoost Confusion Matrix

	Predicted 0	Predicted 1
Actual 0	8023	32
Actual 1	26	7849

After tuning, SVM surprisingly became a significantly stronger predictor, even besting XGBoost in some areas. This is more clear when looking at the ROC and Precision-Recall curves shown below:

TABLE XII
OPTIMIZED XGBOOST CLASSIFICATION REPORT

Class	Precision	Recall	F1-Score	Support
0.0	1.00	1.00	1.00	8055
1.0	1.00	1.00	1.00	7875
Accuracy			1.00	15930
Macro Avg	1.00	1.00	1.00	15930
Weighted Avg	1.00	1.00	1.00	15930

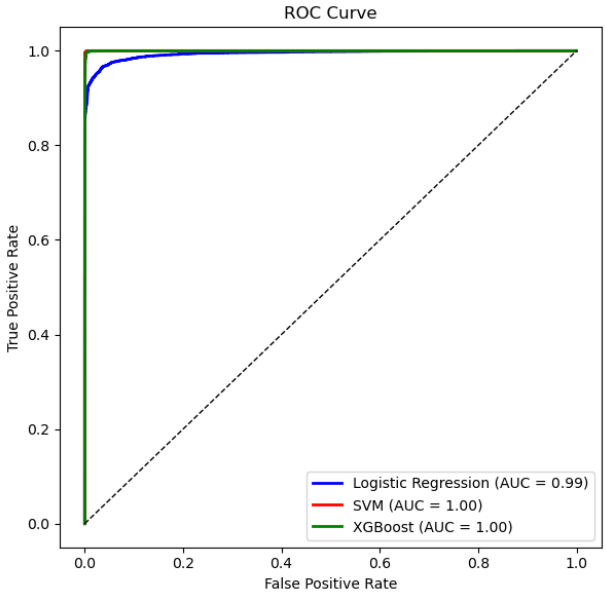

Fig. 3. The ROC Curve for all models after tuning.

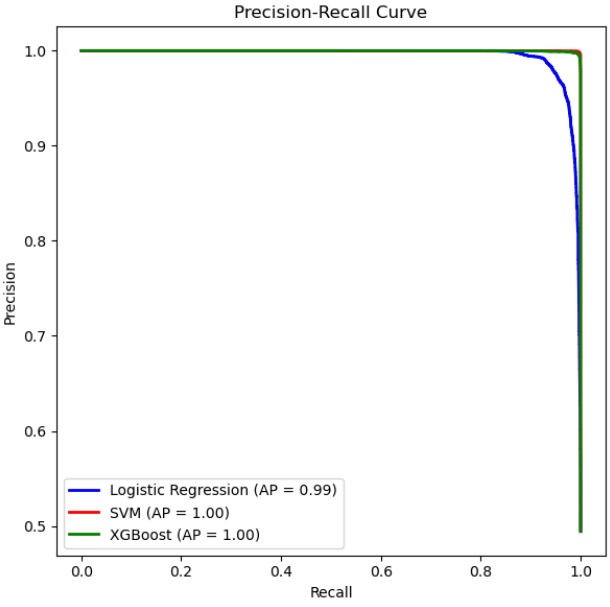

Fig. 4. The Precision-Recall Curve for all models after tuning.

Though its a bit hard to see since they are neck-and-neck, the SVM curve just barely edges out XGBoost as it is slightly higher on both graphs.

VI. CONCLUSIONS

In our study, we assessed the performance of machine learning models for detecting fraudulent credit card transactions using a modern 2023 dataset. We found that XGBoost outperformed other models, achieving near-perfect accuracy in detecting fraudulent transactions. Despite a slight reduction in performance due to hyperparameter tuning, it still exhibited strong results. Logistic regression and SVM also demonstrated solid performance, confirming their relevance for fraud detection tasks.

Our results align with the work of M. Chen (2023), who similarly found that XGBoost outperformed logistic regression, SVM, and random forest models in detecting fraud, achieving nearly perfect accuracy [1]. R. Cherkaoui and E. En-Naimi (2023) also investigated SVM, random forest, and KNN on a similar dataset [2]. Their random forest model achieved the highest precision, but their models shared the same accuracy level. Our study extends their work by incorporating advanced hyperparameter tuning techniques to improve model performance across different algorithms.

Additionally, Zhou et al. (2023) [3] explored methods to handle data imbalance in fraud detection, using techniques such as boundary reconstruction and integrated classification. In contrast, our study sidestepped these strategies by using a balanced dataset from the outset. This choice allowed us to focus on developing models better suited to real-world scenarios without the need for complex balancing techniques.

In summary, our study builds on the findings of these previous works while advancing the field by using a balanced, up-to-date dataset, fine-tuning hyperparameters, and improving fraud detection models through XGBoost and other well-established techniques.

REFERENCES

[1] M. Chen, “Credit Card Fraud Detection Based on Multiple Machine Learning Models: Proceedings of the 2022 6th International Conference on Electronic Information Technology and Computer Engineering.” ACM Other Conferences, Association for Computing Machinery, 15 Mar. 2023, dl.acm.org/doi/10.1145/3573428.3573745.

[2] W. Zhou, X. Xue, and D. Luo, “Credit Card Fraud Detection Using Boundary Reconstruction and Integrated Classification: Proceedings of the 4th International Conference on Big Data Engineering.” ACM Other Conferences, Association for Computing Machinery, 19 July 2022, dl.acm.org/doi/10.1145/3538950.3538962.

[3] R. Cherkaoui and E. En-Naimi “A Comparison of Machine Learning Algorithms for Credit Card Fraud Detection: Proceedings of the 6th International Conference on Networking, Intelligent Systems & Security.” ACM Other Conferences, Association for Computing Machinery, 13 Nov. 2023, dl.acm.org/doi/10.1145/3607720.3607759.

[4] Kaggle, 2023, https://www.kaggle.com/datasets/nelgiriyewithana/credit-card-fraud-detection-dataset-2023