# Credit Card Fraud Detection

NAHEEM NOAH* and DAMILARE OLANIYAN*, University of Denver, USA

## 1 INTRODUCTION

Credit card fraud poses a substantial and ever-growing threat to both individuals and businesses in today's dynamic financial landscape [4]. The perpetrators of credit card fraud employ a myriad of sophisticated techniques to execute their schemes. In its simplest terms, credit card fraud occurs when an individual illicitly uses another person's credit card for personal gain, all the while keeping both the card owner and issuer oblivious to the unauthorized transactions [5]. Importantly, the fraudster has no affiliations with the cardholder or the issuing institution and harbors no intention of either contacting the card owner or assuming responsibility for the incurred charges. While the pilfering of lost or stolen cards remains the most prevalent form of fraud, other insidious methods include identity theft, skimming, counterfeit cards, and mail intercept fraud, among others [4]. A comprehensive overview of these modes of operation and their respective frequencies is shown in Table 1, providing a detailed breakdown for a better understanding of the landscape.

According to the Federal Trade Commission (FTC), credit card fraud was ranked as the second common type of identity theft in 2021 behind fraud involving government benefits or documents but emerged as the most reported type of identity theft in 2022, comprising a staggering 441, 822 reported cases [1]. The financial repercussions stemming from credit card transactions reached a staggering $181M, surpassing losses associated with debit cards, which amounted to $140M [2]. This underscores the pervasive and costly nature of credit card fraud within the financial industry, where unauthorized transactions are conducted using stolen or counterfeit credit card information. It is important for credit card companies to enhance their capabilities in recognizing and thwarting fraudulent transactions, safeguarding customers from being unjustly charged for items they did not acquire.

Linda et al.(2009) emphasize the critical role of businesses and financial institutions, including banks, in proactively preventing and efficiently addressing credit card fraud. The vulnerability of credit card fraud lies in its ease of execution, enabling fraudsters to withdraw substantial amounts without the card owner's knowledge within a short timeframe. The audacity of these criminals is further magnified by their adeptness at camouflaging fraudulent transactions as legitimate ones, rendering fraud detection an intricate and formidable challenge. In light of these complexities, it becomes important for businesses and financial institutions to implement robust strategies and technologies to detect and prevent fraud, thereby fortifying the integrity of the financial ecosystem.

## 2 RELATED WORK

The use of machine learning for the detection of malicious activities has become pivotal across various domains. Machine learning techniques are adept at analyzing large datasets, identifying patterns, and making predictions, making them valuable tools for proactive detection. In cybersecurity, machine learning is employed to detect malware [8] and phishing attacks [14]. In finance, machine learning models contribute to fraud detection by discerning abnormal patterns in transactions [2]. In healthcare, these techniques aid in identifying anomalies in medical data that may indicate fraudulent insurance claims or potential health issues [9].

*Both authors contributed equally to this research.

[1] https://www.ftc.gov/system/files/ftc_gov/pdf/CSN-Data-Book-2022.pdf

[2] https://www.bankrate.com/finance/credit-cards/credit-card-fraud-statistics/#fraud

Authors' address: Naheem Noah; Damilare Olaniyan, University of Denver, USA.

| Method | Percentage |
|---|---|
| Lost or stolen card | 48 |
| Identity theft | 15 |
| Skimming (or cloning) | 14 |
| Counterfeit card | 12 |
| Mail intercept fraud | 6 |
| Other | 5 |

Table 1. Methods of Credit Card Fraud and their percentage of occurrence [4]

In the domain of credit card fraud detection using machine learning has garnered substantial attention in recent years, reflecting the escalating need for robust security measures in the financial sector. Numerous studies have contributed to advancing the understanding and implementation of machine learning techniques for fraud detection. Awoyemi et al. (2017) conducted an in-depth investigation into the performance of Naïve Bayes, k-Nearest Neighbor, and Logistic Regression models on a highly imbalanced credit card fraud dataset sourced from European cardholders, encompassing a substantial 284, 807 transactions. Employing a hybrid technique involving undersampling and oversampling to address the skewed data, the study rigorously evaluated the models based on accuracy, sensitivity, specificity, precision, Matthews correlation coefficient, and balanced classification rate. Intriguingly, the results unveiled optimal accuracy rates for Naive Bayes (97.92%), k-Nearest Neighbor (97.69%), and Logistic Regression (54.86%) [1]. Similarly, Varmedja et al. (2019) undertook a comparative analysis of Logistic Regression, Random Forest, Naive Bayes, and Multilayer Perceptron models on a credit card fraud detection dataset obtained from Kaggle, reflecting transactions made by European cardholders over two days. Given the dataset's inherent imbalance, the researchers applied the SMOTE technique for oversampling. The results showcased the efficacy of each algorithm in credit card fraud detection, with the Random Forest algorithm demonstrating superior performance, boasting a precision rate of 96.38%, a recall of 81.63%, and an outstanding accuracy of 99.96% [20].

Furthermore, the work of Yee et al. (2018) [22] combined machine learning and data mining techniques to discern genuine and non-genuine credit card transactions. The study utilized supervised classification with Bayesian network classifiers, including K2, Tree Augmented Naive Bayes (TAN), Naive Bayes, and J48 classifiers. Prior to classification, the dataset underwent meticulous preprocessing involving normalization and Principal Component Analysis (PCA). Remarkably, all classifiers demonstrated accuracy rates surpassing 95.0%, showcasing the efficacy of the applied techniques. The evaluation process harnessed the power of WEKA, a prominent data mining and machine learning tool, to assess the classifiers' performances. The study conducted evaluations on two datasets, the first being a dummy dataset simulating credit card characteristics, and the second, a dataset transformed using data normalization and PCA techniques. Notably, all Bayesian classifiers exhibited markedly improved results when fed with the filtered data [22]. Taking it forward to real-time fraud detection, Thennakoon et al. (2019). proposed a credit-card fraud detection system by leveraging machine learning algorithms, specifically Logistic Regression, Naive Bayes, K-Nearest Neighbor, and Support Vector Machine. Notably, the study achieved accuracy rates of 74%, 83%, 72%, and an impressive 91% for Logistic Regression, Naive Bayes, K-Nearest Neighbor, and Support Vector Machine, respectively [19]. The incorporation of predictive analytics and an API module further distinguished this system, enabling instantaneous notification to end users via a user-friendly graphical interface the moment a fraudulent transaction occurred. These studies

collectively contribute valuable insights into the landscape of machine learning for fraud detection, highlighting the significance of algorithmic selection and dataset preprocessing techniques.

Extending the foundation laid by prior research, our work focuses on credit card fraud detection, employing machine learning algorithms. Specifically, our study incorporates Logistics Regression (LR), Random Forest Classifier (RFC), Naive Bayes (NB), Stochastic Gradient Descent (SGD), and Support Vector Machine (SVM). Each is chosen for its unique strengths in discerning complex patterns within credit card transaction data. Drawing inspiration from the work of Awoyemi et al.(2017), Varmedja et al.(2019), and Yee et al.(2018), we aim to not only measure but also comprehensively compare the performance of these algorithms [1, 21, 22]. Our evaluation metrics encompass pivotal indicators of classification accuracy, including, Recall, F1 score, False Positive Rate (FPR), and False Negative Rate (FNR) offering a clear understanding of each algorithm's efficacy in detecting fraudulent transactions. By building upon and synthesizing insights from these diverse studies, our research aspires to contribute a comprehensive analysis of machine learning algorithms in the world of credit card fraud detection, with the ultimate goal of enhancing the robustness of fraud detection systems.

## 3 METHODOLOGY

In our pursuit of accurate and consistent credit card fraud detection, we leveraged the anonymized credit card transaction dataset available on Kaggle [3]. This dataset captures transactions conducted by European cardholders in September 2013 over a span of two days, comprising a staggering 284, 807 transactions, with 492 instances of fraud cases. The challenge lies in the highly unbalanced nature of the dataset, where fraudulent transactions constitute a mere 0.172% of the entire dataset. Notably, the dataset primarily consists of numerical input variables resulting from Principal Component Analysis (PCA) transformation, with "Time" and "Amount" being the only features untouched by PCA. Given the dataset's imbalance, we employ the Synthetic Minority Oversampling Technique (SMOTE)[4], a targeted strategy designed to address imbalances by generating synthetic samples for the minority class. Our implementation followed through the steps provided by Yong Min Jia in Kaggle [5]. The data pre-processing and model evaluation process are discussed below.

### 3.1 Data Preprocessing

As a part of data preprocessing, we strategically harnessed the capabilities of *StratifiedShuffleSplit*, a pivotal cross-validator within scikit-learn's model selection module which ensures a harmonious distribution of the target variable across both training and test sets—an imperative consideration, especially when confronted with the complexities of imbalanced datasets exhibiting uneven class distributions. We performed an initial split, earmarking 80% for training and reserving 20% for testing. Within this training subset, an additional 20% was set aside to serve as validation data. We performed an exploration to identify outliers within the dataset visualizing using boxplots for each column, particularly within the "Amount" values and we created a function to filter out the outliers as shown in Figure 2. Furthermore, an astute observation highlighted the pronounced variability in the "Amount" and "Time" columns, prompting the application of feature scaling using *ColumnTransformer* within scikit-learn's compose module, to ensure that these variables, with their high ranges, did not disproportionately influence the model's decision-making process. The transformative process of feature scaling was applied to the training, validating, and testing data. Machine learning models including LR, RFC, NB, SGD, and SVM were trained on our data. Due to the

[3] https://www.kaggle.com/code/ymingj/credit-card-fraud-detection-withscikit-learn/input?select=creditcard.csv
[4] https://www.analyticsvidhya.com/blog/2020/10/overcoming-class-imbalance-using-smote-techniques/
[5] https://www.kaggle.com/code/ymingj/credit-card-fraud-detection-with-scikit-learn/notebook

challenges posed by the dataset's imbalanced nature, we decided to abstain from the conventional metric of the Accuracy Score. This is because accuracy is not sensitive to class imbalance and may provide a falsely optimistic view of a model's performance. Instead, our focus honed in on the metrics of Recall and F1 score, providing a nuanced evaluation of model performance in the face of imbalanced class distribution. By focusing on Recall and F1 scores, we prioritize the model's ability to correctly identify instances of the minority class and minimize false negatives.

### 3.2 Machine Learning Models

*3.2.1 Logistics Regression:* This is a statistical method widely used for binary classification tasks, where the outcome variable has two possible classes. The fundamental idea behind Logistic Regression is to model the probability that a given input belongs to a particular class using the logistic function (also known as the sigmoid function) [11]. The logistic function transforms any real-valued number into a value between 0 and 1, representing a probability. The model then assigns a threshold (usually 0.5), and if the predicted probability is above this threshold, the input is classified into one class; otherwise, it is classified into the other class.

*3.2.2 Random Forest Classifier:* This is an ensemble learning method that operates by constructing a multitude of decision trees during training and outputs the mode of the classes for classification problems or the average prediction for regression tasks [15]. Each decision tree in the forest is constructed using a subset of the training data and a random subset of features. This randomness helps to de-correlate the individual trees, reducing the risk of overfitting and enhancing the model's generalization performance. During the classification process, each tree independently assigns a class, and the final prediction is determined by a majority vote (for classification) or an average (for regression) across all the trees in the forest.

*3.2.3 Naive Bayes:* This is a probabilistic machine learning algorithm based on Bayes' theorem. Despite its simplicity, it is particularly effective for classification tasks, especially in natural language processing and text categorization [16]. The algorithm calculates the probability of each class for a given set of features and assigns the class with the highest probability as the prediction. Naive Bayes is computationally efficient, requiring a relatively small amount of training data to estimate the parameters accurately. It is widely used in situations where the independence assumption does not significantly impact performance.

*3.2.4 Stochastic Gradient Descent:* SGD is a powerful optimization algorithm widely used in machine learning for training models, especially in large-scale and online learning scenarios. Unlike traditional Gradient Descent, which computes the gradient of the entire dataset, SGD updates model parameters iteratively and randomly selects a single data point or a small batch for each update [10]. This randomness introduces a stochastic element that helps the algorithm escape local minima and navigate the loss landscape more efficiently. With each iteration, the algorithm adjusts the model parameters in the opposite direction of the gradient of the loss function, gradually minimizing the overall loss.

*3.2.5 Support Vector Machine:* SVM is a powerful supervised machine learning algorithm designed for both classification and regression tasks. SVM operates by finding the optimal hyperplane that maximally separates data points belonging to different classes in the feature space. This hyperplane is chosen to have the maximum margin, defined as the distance between the nearest data points of each class to the hyperplane [18]. SVM can handle linear and non-linear relationships between features by employing kernel functions, which implicitly map the input data into higher-dimensional spaces. The algorithm aims to classify new instances by placing them in the appropriate side of the hyperplane, with each side corresponding to a distinct class.

### 3.3 Model Evaluation Metrics

*3.3.1 False Positive Rate:* FPR is the ratio of incorrectly predicted positive instances (fraudulent transactions in the context of credit card fraud detection) to the total number of actual negative instances (legitimate transactions). A low FPR is desirable, as it indicates a low rate of false alarms or wrongly identifying legitimate transactions as fraudulent. In other words, it is a type I error or a "false alarm.". The False Positive Rate is calculated as the ratio of false positives to the sum of true negatives and false positives.

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

Where; FP = False Positives and TN = True Negatives

*3.3.2 False Negative Rate:* FNR is the ratio of incorrectly predicted negative instances to the total number of actual positive instances. In credit card fraud detection, this means the rate of failing to identify actual fraudulent transactions. A low FNR is crucial, as it represents a low rate of missing actual instances of fraud. Often referred to as a type II error, a False Negative represents a failure to identify a genuine positive occurrence, and it can have significant consequences in various applications such as fraud detection systems. The False Negative Rate is calculated as the ratio of false negatives to the sum of true positives and false negatives.

$$FNR = \frac{FN}{FN + TP}$$

Where; FN = False Negatives and TP = True Positives

*3.3.3 Recall:* This also known as sensitivity or true positive rate, is a crucial metric in binary and multiclass classification, highlighting a machine learning model's ability to capture all positive instances within a dataset. It is calculated as the ratio of true positives to the sum of true positives and false negatives. Mathematically, recall is expressed as True Positives divided by (True Positives + False Negatives). Recall focuses on minimizing false negatives, making it especially relevant in scenarios where failing to identify positive instances can have serious consequences. A high recall score indicates that the model effectively captures a significant proportion of the actual positive instances, but it should be interpreted alongside other metrics, such as F1 score, as an overemphasis on recall may lead to an increase in false positives.

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

Where; TP = True Positives and FN = False Negatives

*3.3.4 F1 Score:* This is a comprehensive metric in binary and multiclass classification that combines both precision and recall into a single measure, providing a balanced assessment of a machine learning model's performance. It is calculated as the harmonic mean of precision and recall, emphasizing a balance between false positives and false negatives. Mathematically, the F1 score is expressed as 2 times the product of precision and recall, divided by the sum of precision and recall. The F1 score ranges between 0 and 1, where a higher value indicates a better balance between precision and recall. This metric is particularly useful in situations where there is an uneven class distribution or when both false positives and false negatives carry significant consequences. The F1 score is a valuable tool for evaluating and comparing the overall effectiveness of classification models, providing a holistic perspective on their ability to correctly classify positive instances while

minimizing both types of errors.

$$F1\_score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

## 4 RESULT

In the pursuit of identifying the most effective algorithm for detecting fraudulent transactions, we executed a comprehensive evaluation of various machine learning models, scrutinizing their performance across key metrics such as False Positive Rate, False Negative Rate, Recall, and F1 score.

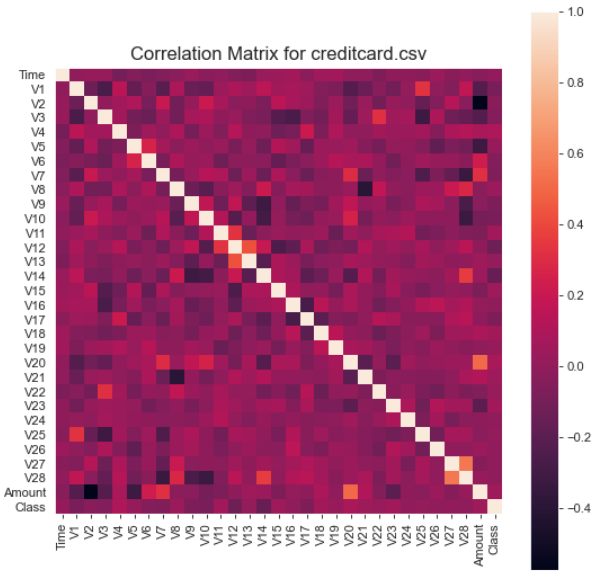


Fig. 1. Heatmap of Correlation

### 4.1 Recall & F1 Score:

Commencing with our baseline model, LR, we observed a Recall of 0.51 and an F1 Score of 0.63. Strikingly similar results were obtained from the SGD Classifier, exhibiting a Recall of 0.52 and an F1 Score of 0.63. These initial algorithms demonstrated relatively lower performance. The SVM, however, showcased substantial improvement with a Recall of 0.60 and an F1 Score of 0.73. The pinnacle of performance was achieved by the RFC, manifesting a Recall of 0.78 and an impressive F1 Score of 0.85. Surprisingly, NB exhibited the highest Recall at 0.84, yet its F1 Score was notably lower at 0.07. A detailed breakdown of these results is shown in Table 2.

### 4.2 False Positive Rate & False Negative Rate:

In terms of FPR and FNR, NB emerges as a paradoxical contender, showcasing the lowest FNR at an astonishingly minute value of 0.0003, indicative of its exceptional ability to minimize missed fraudulent transactions. However, this seemingly impressive feat is juxtaposed against the highest

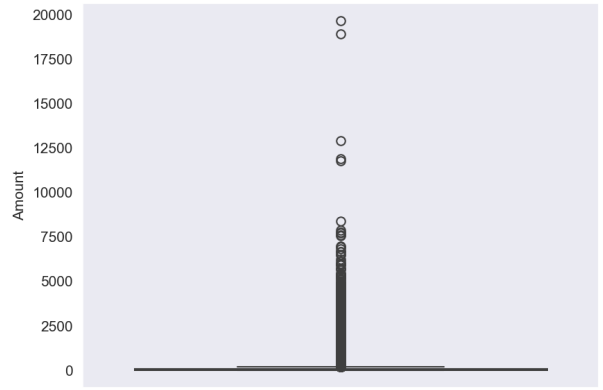


Fig. 2. Outliers in "Amount" column

| Algorithm | Recall | F1 Score | False Positive Rate | False Negative Rate |
|---|---|---|---|---|
| Logistic Regression | 0.51 | 0.63 | 0.16 | 0.0009 |
| SGD | 0.52 | 0.63 | 0.19 | 0.0009 |
| Random Forest | 0.78 | 0.85 | 0.07 | 0.0004 |
| SVM | 0.60 | 0.73 | 0.07 | 0.0007 |
| Naive Bayes | 0.84 | 0.07 | 0.96 | 0.0003 |

Table 2. Evaluation Results of Machine Learning Algorihtms

FPR among the algorithms, standing at 0.96, signaling a substantial propensity for false alarms. On the other hand, RF and SVM align in exhibiting commendably low FPR values at 0.07, with RF further asserting its prowess by attaining a notably lower FNR of 0.0004, in contrast to SVM's 0.0007. LR and SGD share similar FPRs at 0.16 and 0.19, respectively, alongside comparable FNRs at 0.0009. A detailed breakdown of these results is shown in Table 2.

### 4.3 Improving Random Forest Classifier:

The unequivocal selection of RF as our prominent algorithm is grounded in its exemplary performance metrics, boasting the highest Recall and F1 Score, along with the lowest FPR and FNR. In our relentless pursuit of refining the model, we invested in hyperparameter tuning, resulting in optimization of attributes such as *n_estimators* (400), *min_samples_split* (2), *min_samples_leaf* (2), *max_features* (log2), and *bootstrap* (False). This left the FPR and FNR unchanged but elevated the Recall score from 0.78 to 0.80 and the F1 Score from 0.85 to 0.86. Further, we integrated SMOTE in our modeling to address the class imbalance. While this approach didn't yield significant alterations in F1 Score and Recall, it did result in a slightly higher FPR of 0.01. Additionally, introducing the class_weight as "balanced" failed to bring about a discernible shift in outcomes compared to the previous SMOTE resampling. The pinnacle of our Random Forest experimentation was achieved through hyperparameter tuning alone, yielding a Recall score of 0.80, an F1 Score of 0.86, an FPR of 0.07, and an FNR of 0.0004. A detailed breakdown of these results is shown in Table 3. Finally, we compared the results from the different iterations of Random Forest to the unseen dataset—specifically, the test data. Notably, the model exhibited exceptional performance,

surpassing its validation data counterparts across Recall, F1 Score, False Positive Rate, and False Negative Rate. A detailed breakdown of these results is shown in Table 4.

| Random Forest Iteration | Recall | F1 Score | False Positive Rate | False Negative Rate |
|---|---|---|---|---|
| Base model | 0.78 | 0.85 | 0.07 | 0.0004 |
| Base model + hyperparameter tuning | 0.80 | 0.86 | 0.07 | 0.0004 |
| Base model + hyperparameter tuning + resampling | 0.80 | 0.85 | 0.1 | 0.0004 |
| Base model + hyperparameter tuning + resampling + class weight | 0.80 | 0.85 | 0.1 | 0.0004 |

Table 3. Evaluation Results of iterations of Random Forest with Validation Data

| Random Forest Iteration | Recall | F1 Score | False Positive Rate | False Negative Rate |
|---|---|---|---|---|
| Base model | 0.92 | 0.94 | 0.03 | 0.0002 |
| Base model + hyperparameter tuning | 0.91 | 0.94 | 0.02 | 0.0002 |
| Base model + hyperparameter tuning + resampling | 0.94 | 0.95 | 0.03 | 0.0001 |
| Base model + hyperparameter tuning + resampling + class weight | 0.93 | 0.94 | 0.04 | 0.0001 |

Table 4. Evaluation Results of iterations of Random Forest With Test Data

## 5 DISCUSSION

In our analysis, RF emerged as the best performer in the detection of credit card fraud, boasting a remarkable Recall score of 0.80, an F1 Score of 0.86, a low FPR of 0.07, and an impressively minimal FNR of 0.0004. This aligns with the findings of Dornadula and Geetha (2019), who reported superior results for Random Forest, showcasing an Accuracy of 0.9998, Precision of 0.9996, and Matthews Correlation Coefficient (MCC) of 0.9996, surpassing the performance of Isolation Forest, Logistic Regression, and Decision Trees [6]. Similarly, Varmedja et al. (2019) conducted a comparative study involving Logistic Regression, Random Forest, Naive Bayes, and Multilayer Perceptron, where Random Forest outshone the competition with a Precision of 96.38%, Recall of 81.63%, and an impressive Accuracy of 99.96% [21].

Random Forest's superior performance in detecting credit card fraud can be attributed to several key strengths inherent in its algorithmic design. Operating as an ensemble learning method, Random

Forest leverages the collective wisdom of multiple decision trees to enhance predictive accuracy and mitigate overfitting. This ensemble approach proves particularly advantageous in capturing intricate patterns and non-linear relationships inherent in credit card fraud scenarios [3]. The algorithm's capability to assign importance scores to features facilitates a focused attention on relevant aspects, contributing to its efficacy in discerning fraudulent patterns. Furthermore, Random Forest's robustness to overfitting, flexibility in hyperparameter tuning, and effective handling of imbalanced data, where fraud instances are outnumbered, collectively contribute to its standout performance in the challenging task of credit card fraud detection [13].

Hyperparameter tuning, involving the meticulous adjustment of key model parameters, proved to be a more influential factor in elevating the Recall and F1 Score in our case [17]. Fine-tuning parameters such as the number of trees and minimum samples per split enabled the model to better discern intricate patterns within the data. Conversely, the application of SMOTE, designed to address class imbalance by generating synthetic instances of the minority class, did not yield a substantial improvement in the evaluated metrics but performed well with the test data [7]. The outcomes suggest that the model's performance improvement was more driven by optimizing its internal mechanisms than by addressing class distribution imbalance.

In the obtained results, Naive Bayes exhibits a high recall but a low F1 score. This discrepancy often arises from differences in precision and could be attributed to various factors. If the dataset is imbalanced, with a majority class dominating, Naive Bayes may favor the majority class, leading to a high recall for the minority class but a lower precision, resulting in a diminished F1 score [12]. Additionally, violations of the independence assumption, sensitivity to outliers, misclassification of negative instances, and the impact of feature correlation could contribute to this observed pattern. The probabilistic nature of Naive Bayes and the default decision threshold for classification may also play a role.

## 6 CONCLUSION

We conducted a comprehensive investigation into the performance of various machine learning algorithms for credit card fraud detection. The algorithms evaluated included Logistic Regression, Stochastic Gradient Descent, Support Vector Machine, Random Forest, and Naive Bayes. The models were trained and tested on a real-world credit card transaction dataset featuring a high degree of class imbalance, with only 0.172% of transactions being fraudulent. The results reveal Random Forest as the standout top performer, achieving a Recall of 0.80, an F1 Score of 0.86, a low False Positive Rate of 0.07, and an impressively minimal False Negative Rate of 0.0004 after hyperparameter tuning. This stellar performance is attributed to Random Forest's ensemble approach leveraging multiple decision trees, robustness to overfitting, and flexibility in tuning parameters. The findings provide valuable insights into the real-world application of machine learning for the critical task of fraud detection in the financial sector. This research can inform the design of robust fraud detection systems that leverage the predictive strengths of algorithms like Random Forest to identify illicit activities and safeguard customers. Further research could explore combining algorithms in ensembles or testing on larger, more recent datasets. As fraud techniques evolve, it is imperative that detection systems also advance through the thoughtful application of machine learning.

## REFERENCES

[1] John O Awoyemi, Adebayo O Adetunmbi, and Samuel A Oluwadare. Credit card fraud detection using machine learning techniques: A comparative analysis. In *2017 international conference on computing networking and informatics (ICCNI)*, pages 1–9. IEEE, 2017.

[2] Alexander Bakumenko and Ahmed Elragal. Detecting anomalies in financial data using machine learning algorithms. *Systems*, 10(5):130, 2022.

[3] Mariana Belgiu and Lucian Drăguţ. Random forest in remote sensing: A review of applications and future directions. *ISPRS journal of photogrammetry and remote sensing*, 114:24–31, 2016.

[4] Tej Paul Bhatla, Vikram Prabhu, and Amit Dua. Understanding credit card frauds. *Cards business review*, 1(6):1–15, 2003.

[5] Linda Delamaire, Hussein Abdou, and John Pointon. Credit card fraud and detection techniques: a review. *Banks and Bank systems*, 4(2):57–68, 2009.

[6] Vaishnavi Nath Dornadula and Sa Geetha. Credit card fraud detection using machine learning algorithms. *Procedia computer science*, 165:631–641, 2019.

[7] Alberto Fernández, Salvador Garcia, Francisco Herrera, and Nitesh V Chawla. Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *Journal of artificial intelligence research*, 61:863–905, 2018.

[8] Ivan Firdausi, Alva Erwin, Anto Satriyo Nugroho, et al. Analysis of machine learning techniques used in behavior-based malware detection. In *2010 second international conference on advances in computing, control, and telecommunication technologies*, pages 201–203. IEEE, 2010.

[9] Hossein Joudaki, Arash Rashidian, Behrouz Minaei-Bidgoli, Mahmood Mahmoodi, Bijan Geraili, Mahdi Nasiri, and Mohammad Arab. Using data mining to detect health care fraud and abuse: a review of literature. *Global journal of health science*, 7(1):194, 2015.

[10] Nikhil Ketkar and Nikhil Ketkar. Stochastic gradient descent. *Deep learning with Python: A hands-on introduction*, pages 113–132, 2017.

[11] Michael P LaValley. Logistic regression. *Circulation*, 117(18):2395–2399, 2008.

[12] Yang Lu, Yiu-Ming Cheung, and Yuan Yan Tang. Bayes imbalance impact index: A measure of class imbalanced data set for classification problem. *IEEE transactions on neural networks and learning systems*, 31(9):3525–3539, 2019.

[13] AS More and Dipti P Rana. Review of random forest classification techniques to resolve data imbalance. In *2017 1st International Conference on Intelligent Systems and Information Management (ICISIM)*, pages 72–78. IEEE, 2017.

[14] Naheem Noah, Abebe Tayachew, Stuart Ryan, and Sanchari Das. Phishercop: Developing an nlp-based automated tool for phishing detection. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 66, pages 2093–2097. SAGE Publications Sage CA: Los Angeles, CA, 2022.

[15] Mahesh Pal. Random forest classifier for remote sensing classification. *International journal of remote sensing*, 26(1):217–222, 2005.

[16] Irina Rish et al. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46, 2001.

[17] Patrick Schratz, Jannes Muenchow, Eugenia Iturritxa, Jakob Richter, and Alexander Brenning. Hyperparameter tuning and performance assessment of statistical and machine-learning algorithms using spatial data. *Ecological Modelling*, 406:109–120, 2019.

[18] Shan Suthaharan and Shan Suthaharan. Support vector machine. *Machine learning models and algorithms for big data classification: thinking with examples for effective learning*, pages 207–235, 2016.

[19] Anuruddha Thennakoon, Chee Bhagyani, Sasitha Premadasa, Shalitha Mihiranga, and Nuwan Kuruwitaarachchi. Real-time credit card fraud detection using machine learning. In *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pages 488–493. IEEE, 2019.

[20] Dejan Varmedja, Mirjana Karanovic, Srdjan Sladojevic, Marko Arsenovic, and Andras Anderla. Credit card fraud detection - machine learning methods. In *2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH)*, pages 1–5, 2019.

[21] Dejan Varmedja, Mirjana Karanovic, Srdjan Sladojevic, Marko Arsenovic, and Andras Anderla. Credit card fraud detection-machine learning methods. In *2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH)*, pages 1–5. IEEE, 2019.

[22] Ong Shu Yee, Saravanan Sagadevan, and Nurul Hashimah Ahamed Hassain Malim. Credit card fraud detection using machine learning as data mining technique. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 10(1-4):23–27, 2018.