Real-Time Fraud Detection Using Machine Learning

Article *in* Journal of Data Analysis and Information Processing · January 2024

DOI: 10.4236/jdaip.2024.122011

Benjamin Borketey
University of Akron

Journal of Data Analysis and Information Processing, 2024, 12, 189-209
https://www.scirp.org/journal/jdaip
ISSN Online: 2327-7203
ISSN Print: 2327-7211

Real-Time Fraud Detection Using Machine Learning

Benjamin Borketey

Department of Economics, The University of Akron, Akron, Ohio, USA
Email: bborketey9@gmail.com

How to cite this paper: Borketey, B. (2024) Real-Time Fraud Detection Using Machine Learning. *Journal of Data Analysis and Information Processing*, **12**, 189-209.
https://doi.org/10.4236/jdaip.2024.122011

Received: April 4, 2024
Accepted: May 14, 2024
Published: May 17, 2024

Abstract

Credit card fraud remains a significant challenge, with financial losses and consumer protection at stake. This study addresses the need for practical, real-time fraud detection methodologies. Using a Kaggle credit card dataset, I tackle class imbalance using the Synthetic Minority Oversampling Technique (SMOTE) to enhance modeling efficiency. I compare several machine learning algorithms, including Logistic Regression, Linear Discriminant Analysis, K-nearest Neighbors, Classification and Regression Tree, Naive Bayes, Support Vector, Random Forest, XGBoost, and Light Gradient-Boosting Machine to classify transactions as fraud or genuine. Rigorous evaluation metrics, such as AUC, PRAUC, F1, KS, Recall, and Precision, identify the Random Forest as the best performer in detecting fraudulent activities. The Random Forest model successfully identifies approximately 92% of transactions scoring 90 and above as fraudulent, equating to a detection rate of over 70% for all fraudulent transactions in the test dataset. Moreover, the model captures more than half of the fraud in each bin of the test dataset. SHAP values provide model explainability, with the SHAP summary plot highlighting the global importance of individual features, such as "V12" and "V14". SHAP force plots offer local interpretability, revealing the impact of specific features on individual predictions. This study demonstrates the potential of machine learning, particularly the Random Forest model, for real-time credit card fraud detection, offering a promising approach to mitigate financial losses and protect consumers.

Keywords

Credit Card, Fraud Detection, Machine Learning, SHAP Values, Random Forest

1. Introduction

Financial fraud involves misleading or withholding information from victims regarding promised benefits, commodities, or services frequently to obtain economic advantage. In the U.S. financial sector, identity theft is a common theme in various credit card fraud schemes. Identity theft occurs when a fraudster obtains access to or opens an account using the victim's personal information, often by stealing utility bills or bank statements. Scams can happen over the phone or through email, with the con artist posing as a bank official and asking for private information. The criminal then reports a missing card to the cardholder's bank using this personal information, and the bank issues a new card to the criminal.

Credit card fraud occurs through various channels, including online, over the phone (both by text and voice), and in-person [1]. It involves a wide range of illegal activities, such as card skimming, where fraudsters install skimmers on gas pumps, ATMs, and point of sale (POS) systems to steal card information where customers swipe their credit or debit cards [2]. In account takeovers, fraudsters use stolen personal information to contact credit card companies, pretending to be legitimate cardholders [3].

Credit card application fraud involves using stolen personally identifiable information, such as names, addresses, birthdays, and social security numbers, to apply for credit cards in card-not-present (CNP) and card-present transactions. Other forms of credit card fraud include complex online scams and synthetic identity fraud, where criminals create fake identities using a combination of real and fabricated PII. These fraudulent activities result in significant financial losses and emphasize the need for robust fraud detection and prevention measures.

The growing popularity of credit cards has resulted in a rise in online business transactions and the convenience of electronic payment systems. However, this widespread adoption has also given rise to fraudulent activities. According to the Federal Trade Commission's Consumer Sentinel Network data book for 2021 ([4], p. 8), the primary types of fraud reported in the United States were identity theft (25.01%), imposter scams (17.16%), issues with Credit Bureaus and online shopping scams (6.94%). Collectively, these fraudulent activities accounted for 50% of all reported fraud cases in the country.

The financial sector is increasingly alarmed by the escalating threat of credit card theft, which results in the loss of billions of dollars to fraudulent activity annually. The impact of credit card fraud is pervasive, affecting not only the individual consumer but also the issuing banks, businesses, and government agencies. The consumer bears the brunt of financial losses and damage to their credit score due to unauthorized charges on their credit cards, impairing their access to loans and other financial services. Larger-scale credit card fraud inflicts a staggering annual revenue loss of billions of dollars on banks. The Federal Trade Commission's recent data for 2022 revealed that consumers reported a staggering $8.8 billion loss to fraud, a more than 30% increase from the previous year [5]. Furthermore, the FBI's Internet Crime Report for 2022 highlighted that Credit Card/Check Fraud accounted for $264.1 million in reported losses, while Identity Theft led to $189.2 million in losses ([6], p. 29).

Fraud detection involves analyzing customers' transaction behavior to determine the legitimacy of transactions. With the increasing prevalence of electronic transactions, detecting and preventing fraudulent activities has become more challenging. Traditional rule-based approaches adopted by banks and financial institutions often struggle to keep up with evolving fraud techniques and the sophisticated methods employed by fraudsters.

Machine learning has emerged as a powerful and efficient methodology for combating credit card fraud. These systems leverage models trained with historical data on both fraudulent and legitimate activities to autonomously identify characteristic patterns and recognize them when they reoccur. By scrutinizing large volumes of transaction data and pinpointing suspicious patterns, machine learning models can accurately classify transactions as either genuine or fraudulent, providing a robust defense against evolving fraud techniques.

This research aims to identify the most efficient methodology for detecting fraud, identify the most important features using the chosen model, and determine the percentage of fraudulent activities detected (detection rate) by the model in the riskiest bin in real-time. Specifically, this research adds value to the current literature by providing a real-time demonstration of an efficient methodology for detecting and preventing fraud and also provides the percentage of fraudulent activities detected by the model in the riskiest bin. Additionally, SHAP values will be employed to study the global and local influence of each feature on fraud. These objectives are crucial in the ongoing battle against credit card fraud, as they will contribute to the development of more effective and robust fraud detection and prevention measures.

2. Literature Review

The existing literature extensively examines optimal approaches for detecting and preventing financial fraud through machine learning. In [7], four machine learning models: Support Vector Machine (SVM), Naïve Bayes (NB), K-Nearest Neighbor (KNN), and Logistic Regression (LR), were assessed alongside resampling techniques to tackle data imbalance challenges. The study identified the optimal models for addressing fraud. LR (74%), NB (83%), KNN (72%), and SVM (72%) demonstrated the highest accuracy rates in capturing the fraud patterns (unknown site address, ISO response code, suspicious transactions above $100), respectively. In [8], the usage of an artificial neural network (ANN) trained through the simulated annealing (SA) algorithm was optimal in real-time credit card fraud detection compared to alternative models such as decision trees, support vector machines, genetic algorithms, and back propagation. In a different study [9], a supervised-based classification approach was employed, using Bayesian network classifiers, namely K2, Naïve Bayes, logistic, and J48 classifiers[1]. After employing normalization and Principal Component Analysis, all clas-

[1] J48, also known as C4.5, is a decision tree classifier developed by Ross Quinlan for machine learning and data mining tasks. It constructs a tree by recursively partitioning data based on the most informative attributes, using measures like information gain.

sifiers achieved accuracy rates exceeding 95%. Furthermore, research conducted on how to detect fraudulent cash-back transactions in e-commerce platforms in Indonesia [10] compared three supervised classification algorithms: K-Nearest Neighbor (k-NN), Convolutional Neural Networks (CNN), and Long Short-Term Memory (LSTM), along with a combined CNN-LSTM model, to classify transactions as fraudulent or not. The findings indicated that the K-NN algorithm outperformed the other models, achieving an accuracy of 83.82% on the test data, while the CNN-LSTM model demonstrated the lowest accuracy at 52.14%.

Many credit card companies are hesitant to disclose detailed information beyond a basic acknowledgment of breaches, resulting in a significant amount of unreported breach data. This complexity, coupled with legal regulations safeguarding sensitive data, has led to a dearth of research in North America. Consequently, existing literature predominantly focuses on the effectiveness of various machine learning techniques in fraud detection using European datasets, highlighting the importance and complexity of this topic.

In [11], the Kaggle European dataset was used to identify the optimal algorithm for fraud detection. LR achieved an accuracy of 97.46%, NB demonstrated an accuracy of 99.23%, Multilayer Perceptron (MP) achieved 99.93% accuracy, and Random Forest (RF) produced the highest accuracy at 99.96%. The RF and Adaboost algorithms were also applied to the Kaggle credit card fraud dataset, and performance was evaluated based on accuracy, precision, recall, and F1-score [12]. The RF algorithm had the highest accuracy, precision, recall, and F1 score among the two algorithms. It also had the lowest false positive and false negative rate. Further, the ROC curve showed that the RF algorithm had a high area under the curve, indicating a good trade-off between sensitivity and specificity. On the contrary, while Random Forest demonstrates decent AUC and MCC metrics, it carries a higher failure cost than SVM, KNN, and CNN in [13] on the European dataset.

Furthermore, in [14], various machine-learning techniques for detecting credit card fraud were explored on the European dataset. Accuracy, precision, and Matthews Correlation Coefficient were used as evaluation metrics. After addressing the class imbalance using the SMOTE technique, the results highlighted logistic regression, decision tree, and random forest as top performers. A hybrid sampling technique was also implemented to address data imbalance by oversampling the positive class (fraud) and undersampling the negative class [15] within the European dataset. The findings reveal that NB achieved the highest accuracy (97.92%), followed by KNN (97.69%) and LR (54.86%), respectively. Additionally, KNN outperformed NB and LR in specificity, precision, and MCC, demonstrating superior performance on the sampled data.

While the previous literature [11] [12] [13] [14], and [15] focused on the application of Machine Learning to detect and prevent credit card fraud using European datasets, none illustrated the practical application of the models for real-time fraud detection. This research adds value to the current literature by addressing these gaps and providing a real-time demonstration of an efficient methodology for detecting and preventing fraud on the European data set from Kaggle.

There is a growing consensus on the necessity for real-time fraud detection [7] [8] [9] [16] [17] [18]. Although numerous studies in the literature discuss real-time fraud detection using machine learning models, none of them explicitly demonstrate the quantitative effectiveness of these models, such as the amount or percentage of fraud detected in real-time. For example, in [18], a live credit card fraud detection system is introduced, utilizing deep neural network technology with an auto-encoder. The approach involves two phases: periodic offline training on historical data for model construction and real-time prediction of new data using the established model. On the European dataset, these approaches were compared to four other binary classification methods—linear SVM, LR, non-linear auto-regression NN, and NN classification. Though the final results indicate that the deep NN with auto-encoder achieved the highest F1 score and precision, closely followed by LR in accuracy and recall, the percentage of fraud detected by the models on the European data sets was not discussed.

To address this gap in the existing literature, I determine the percentage of fraudulent transactions detected by the optimal model in the riskiest bin in real-time.

With the increasing adoption of ML models, there has equally been growing consensus on the inherent challenges of explainability in complex and black-box[2] models, such as deep neural networks and ensemble methods. Several techniques and frameworks have been proposed to address these challenges, including Local Interpretable Model-agnostic Explanations (LIME) and model-specific interpretability methods. However, none of the studies on the European data set aforementioned have explored the use of SHAP values in interpreting fraud. The final part of the study focuses on leveraging SHAP values to assess the contribution of each variable to fraud detection.

3. Methodology

The primary research methodology involves comparing various machine learning approaches logistic regression (LR), Linear Discriminant Analysis (LDA), K-nearest Neighbors (KNN), Classification and Regression Tree (CART), Naive Bayes (NB), Support Vector Machine (SVM), Random Forest (RF), XGBoost (XGB), Light Gradient-Boosting Machine (LightGBM) and selecting the best optimal model based on evaluation metrics such as AUC, PRAUC, F1, KS, Recall, and Precision. The chosen model will then be employed in subsequent analyses to determine how features contribute to and explain fraud detection using feature importance scores and SHAP values.

3.1. Data Processing and Feature Selection

The research used a credit card dataset sourced from Kaggle [19], which consists

[2] Black-box models are pre-packaged ML algorithms such as Gradient Boosting etc. Unlike the traditional ML Models like Logistic regression, black-box models are opaque and non-transparent.

of transactions conducted by European cardholders over two days in September 2013. The dataset presents transactions that occurred in two days with 492 frauds out of 284,807 transactions. As is customary in fraud datasets, non-fraudulent transactions vastly outnumber fraudulent ones, with 284,315 (99.83%) non-fraudulent transactions and 492 (0.17%) fraudulent transactions. To protect the confidentiality of customer features, Principal Component Analysis (PCA) transformation was applied to the original dataset, excluding identifiable information features such as "time" and "amount." Thus features V1, V2…V28 are the principal components obtained with PCA. The "Time" feature contains the seconds elapsed between each transaction and the first transaction in the dataset. The "Amount" feature is the transaction amount. The target "Class" is the response variable and it takes the value of 1 in case of fraud and 0 for non-fraud.

To ensure accurate modeling, one of any two features with a correlation coefficient of 0.99 is excluded from the model training process. To address the class imbalance [11] [17] [20] used the highly effective synthetic minority oversampling technique. This technique tackles class imbalance in machine learning by generating synthetic samples for the minority class. It identifies the minority class instances, selects their nearest neighbors, and creates new artificial examples along the lines connecting these neighbors, effectively expanding the dataset with realistic representations of the underrepresented class. To address the class imbalance for efficient modeling, the Synthetic Minority Oversampling Technique (SMOTE) is applied to the minority class (non-fraud). This results in a final balanced dataset comprising 199,002 instances of fraud and 199,002 instances of non-fraud.

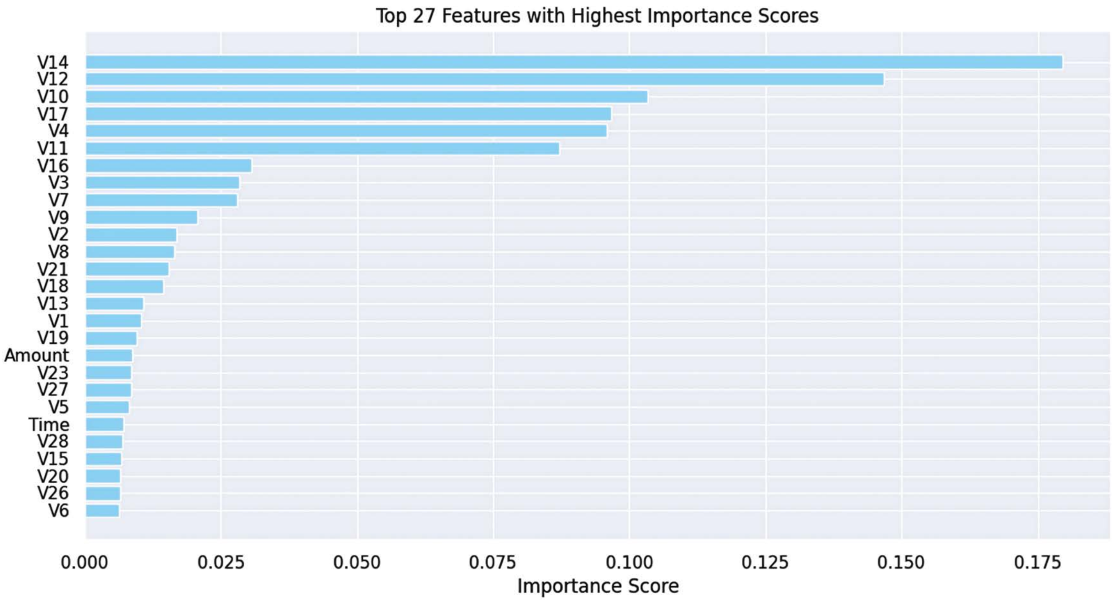

Figure 1. Top 27 features with highest importance score.

Random Forest feature importance is used to select the top 27 features in the early run. This selection aims to achieve faster training times, prevent overfitting, and improve overall model predictions. Figure 1 shows the top 27 features selected using the Random Forest importance metric. Subsequently, the final dataset, containing only the selected features, is divided into a 70% train set and a 30% test set for further analysis.

3.2. Model Building

The primary research methodology involves comparing machine learning approaches and selecting the best model based on evaluation metrics. An automatic algorithm with 10-fold cross-validation is implemented in the building process to achieve this goal. This process comprises the following models:

Logistic Regression (LR) is a linear classification model that plays a crucial role in binary classification tasks. It predicts the relationship between a dependent variable and one or more independent variables. This model leverages the logistic function, the sigmoid function, to transform a linear combination of independent variables into a probability score.

LR offers interpretability, enabling an easy understanding of the impact of independent variables on outcomes, and allows probability estimation for assessing prediction uncertainty. It demonstrates robustness to noise and irrelevant features, ensuring effectiveness in high-dimensional datasets, and boasts computational efficiency, making it suitable for large-scale applications. However, LR assumes a linear relationship between variables, limiting its expressiveness in capturing complex patterns, and it can be sensitive to outliers, potentially skewing predictions. Additionally, LR is primarily designed for binary classification, requiring modifications for multi-class tasks, and it assumes independence among observations, which might not hold in all scenarios.

Linear Discriminant Analysis (LDA): This model predicts the class of the dependent variable by using the linear combination of the independent variables. LDA aims to maximize class separation while minimizing within-class variance by identifying a linear combination of independent variables. Discriminant functions derived from this process are then used to classify new observations using a designated decision rule. Additionally, LDA can reduce dimensionality by projecting data into a lower-dimensional space while preserving the distinct separation between classes.

LDA offers the advantage of dimensionality reduction by projecting data into a lower-dimensional space while preserving distinct class separation. However, LDA can be sensitive to outliers and assumes that independent variables are normally distributed within each class, potentially leading to biased parameter estimates.

K-nearest Neighbors (KNN): KNN, a versatile algorithm, uses proximity to make classifications or predictions about the grouping of an individual data point. It is applicable to both classification and regression problems, assessing similarity by considering the k-nearest neighbors in the feature space. In classification, KNN assigns a data point to the majority class among its neighbors, while in regression, it calculates the average of their values. This algorithm proves effective for diverse predictive tasks, adapting its approach based on the specific requirements of the problem at hand, giving you a wide range of options to choose from.

KNN's simplicity and adaptability make it accessible to beginners and suitable for diverse predictive tasks. However, KNN's computational complexity and memory intensity can pose challenges, especially with large datasets, and its predictions may be sensitive to noise and outliers. Additionally, selecting an optimal value for the parameter k is crucial for achieving optimal performance and often requires a great amount of time for experimenting and tuning.

Classification and Regression Tree (CART): CART, a widely used algorithm for predictive modeling and decision-making, can be employed for both classification and regression tasks. It provides a tree-like structure that represents decision rules and splits in the data. In classification, the tree categorizes instances into different classes, while in regression, it predicts numerical values. The interpretability of CART makes it a powerful tool, giving you confidence in your understanding of the model's decision-making process.

CART's interpretability is a significant advantage, it provides transparency into the model's decision-making process, which instills confidence in its outcomes. However, CART also has limitations. While it is easy to interpret and visualize, it can be prone to overfitting, particularly when the tree depth is not adequately controlled. Additionally, CART may lack robustness when faced with small variations in the data, potentially leading to unstable predictions.

Naive Bayes (NB): Naive Bayes, a computationally efficient and straightforward algorithm, uses Bayes' theorem to assign a probability to every possible value in the target class, and the resulting distribution is then condensed into a single prediction. It calculates the likelihood of each class based on observed data and combines it with prior probabilities for making predictions. NB offers quick and effective predictions. Its efficiency provides reassurance about performance, particularly with large datasets.

However, NB assumes feature independence, which may not always hold in real-world scenarios. This assumption can limit its ability to capture complex relationships between features, potentially leading to suboptimal performance in certain cases.

Support Vector Machine (SVM): SVM finds a hyperplane that best fits the data points in a continuous space instead of fitting a line to the data points. It can be used in regression and classification tasks. SVM aims to find the hyperplane that maximizes the margin between different classes. While versatile enough for regression and classification tasks, SVM excels particularly in solving classification problems. Its ability to handle complex data and find non-linear decision boundaries makes SVM a powerful tool in various fields of machine learning. SVM's strengths lie in its effectiveness in high-dimensional spaces and its robustness to overfitting, especially when using appropriate regularization.

However, SVM's computational complexity increases with the size of the dataset, making it less suitable for large-scale applications. Additionally, SVM's performance heavily depends on the choice of kernel function and its associated parameters and requires careful tuning to achieve optimal results.

Random Forest (RF): Random Forest involves the creation of multiple decision trees, each constructed using distinct random subsets of the data and its features. Each decision tree functions as an individual "expert," offering its perspective on the data classification. To make predictions, the algorithm computes predictions from each decision tree and ultimately selects the most frequently occurring outcome among these individual results.

RF boasts high accuracy due to its ability to reduce overfitting and handle noise effectively. It is versatile, accommodating different types of data, and provides insights into feature importance. However, RF models can be complex and computationally expensive to train, especially with large datasets. Additionally, they may exhibit a bias towards majority classes in imbalanced datasets. Despite these challenges, Random Forest remains widely used and valued for its robustness, accuracy, and versatility in classification tasks.

XGBoost (XGB): XGBoost is particularly popular in various data science and machine learning competitions on platforms like Kaggle due to its high predictive accuracy and versatility. It is designed for classification and regression tasks and is known for its efficiency, scalability, and ability to handle complex structured data. Its success in handling a wide range of datasets and delivering robust performance has made XGBoost a go-to choice for analysts seeking superior predictive models in various applications. XGB advantages include high predictive accuracy, efficient computational performance, scalability to large datasets, and the ability to handle complex structured data.

However, XGBoost also has its limitations. It can be computationally expensive, especially when dealing with large datasets and complex models. Additionally, XGBoost's performance heavily depends on hyperparameter tuning which requires careful optimization to achieve optimal results.

Light Gradient-Boosting Machine (LightGBM): LightGBM is a fast, distributed, high-performance gradient-boosting framework based on decision tree algorithms. It is used for ranking, classification, and many other machine-learning tasks. Its capability to handle large datasets and deliver quick, accurate results makes LightGBM particularly well-suited for applications where speed and performance are crucial, solidifying its popularity in the machine-learning community.

However, LightGBM's performance can be sensitive to hyperparameters, necessitating careful tuning for optimal results. Additionally, its inner workings may pose complexity, demanding a deeper understanding for effective utilization, especially among users less familiar with gradient boosting and decision tree algorithms. While scalability challenges may arise in setting up distributed training environments, and there's a risk of overfitting with high-capacity models.

3.3. Model Selection Metrics

Due to the class imbalance, accuracy may not be the most suitable metric for performance evaluation [11]. Instead, additional metrics, including AUC, F1 score, Precision, and Recall, were employed in evaluating model performance. To ensure a thorough evaluation of the model's effectiveness in handling class imbalance, PRUAC was incorporated. KS was also included, which quantifies the maximum separation between the cumulative distribution of fraud and non-fraud instances. Descriptions of these performance metrics are provided below.

$\text{Accuracy} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}}$: This metric measures the number of correct predictions made by a model in relation to the total number of predictions made. The range $\in [0, 1]$.

$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}}$: Precision calculates the ratio of correctly classified fraud transactions to all transactions classified as fraud. The range $\in [0, 1]$.

$\text{Recall or TPR} = \frac{\text{TP}}{\text{TP}+\text{FN}}$: This measures the ratio of correctly classified fraud transactions to all actual fraudulent transactions. The range $\in [0, 1]$.

$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$: This combines the precision and recall using the harmonic mean. It provides a balanced measure of a model's performance. The range $\in [0, 1]$.

$\text{FPR} = \frac{\text{FP}}{\text{FP}+\text{TN}}$: This measures the proportion of non-fraud transactions that are incorrectly classified as fraudulent transaction. The range $\in [0, 1]$.

Where:

TP is the number of transactions correctly classified as fraud.

TN is the number of transactions correctly classified as non-fraud.

FN is the number of fraud transactions wrongly classified as non-fraud.

FP is the number of non-fraud transaction wrongly classified as fraud

KS: The Kolmogorov Smirnov test (KS) measures the maximum separation between fraudulent and non-fraudulent transaction distribution, which is in the range $\in [0, 1]$.

AUC: This metric summarizes the trade-off between a classifier's true and false favorable rates. It quantifies a classifier's ability to distinguish between positive and negative classes—the range $\in [0, 1]$.

PRAUC: PRAUC summarizes the precision-recall trade-off across different classification thresholds. It calculates the area under the precision-recall curve, which plots precision against recall. A high PRAUC indicates a model that maintains high precision while achieving high recall. This metric is often used for fraud detection, anomaly detection, and imbalanced classification problems—the range ∈ [0, 1].

4. Results and Discussion

4.1. Model Performance in Train Data

The Random Forest model emerged as the best-performing model, achieving the highest KS score of 99.99% and an AUC of 99.99%, demonstrating its robust ability to effectively distinguish between fraudulent and non-fraudulent transactions in Table 1. Similar results were obtained in [1] [14] [19] where the RF produced the best accuracy of 99.96% from among a list of classifiers on the European data set.

Moreover, the Random Forest model produced the highest accuracy rate of 99.99%, along with the highest precision and recall rates of 99.98% and 99.99%, respectively. This high F1 score of 99.99% signifies a well-balanced trade-off between precise positive predictions (precision) and the comprehensive capture of positive instances (recall) by the RF model. Furthermore, the PRAUC value of 99.99% obtained from the Random Forest model shows its superior ability to differentiate between positive and negative classes compared to all other models. See Table 1.

Table 1. Model performance metric on train data sets.

Model	KS	AUC	F1-Score	Recall	PRAUC	Precision	Accuracy
LR	0.8937	0.9904	0.9405	0.9125	0.9912	0.9702	0.9422
LDA	0.8710	0.9774	0.9080	0.8437	0.9784	0.9828	0.9145
KNN	0.9993	0.9997	0.9989	1.0000	0.9993	0.9979	0.9989
CART	0.9965	0.9983	0.9983	0.9991	0.9970	0.9974	0.9983
NB	0.8262	0.9501	0.9063	0.8487	0.9542	0.9722	0.9122
SVM	0.9675	0.9984	0.9836	0.9862	0.9982	0.9810	0.9836
RF	0.9999	0.9999	0.9999	0.9999	0.9999	0.9998	0.9999
XGB	0.9980	0.9999	0.9989	0.9998	0.9999	0.9979	0.9989
Light GBM	0.9992	0.9999	0.9993	0.9999	0.9999	0.9986	0.9993

4.2. Detection of Overfitting (Performance on Test Set)

Assessing the performance of the models on the test data is crucial because it provides an unbiased evaluation of how well each model generalizes to unseen data. This evaluation ensures that the models have not merely memorized the training data (a phenomenon known as overfitting) but can accurately predict new, real-world examples. Evaluating performance on a separate, unseen dataset helps determine the model's reliability and potential to perform well in practical applications, aiding in selecting the best model.

The fitted models are employed to predict the test dataset, and the performance results from the test dataset are compared to those obtained in the training data to detect the possibility of overfitting. Overfitting occurs when a model performs exceptionally well on the training data but fails to generalize effectively to new, unseen data.

Among the models, the Random Forest (RF) model exhibits the lowest reduction in performance metrics when transitioning from the training dataset to the test dataset. A comparison of performance metrics in the training data, as shown in Table 1, to those in the test data, as presented in Table 2, reveals a reduction of less than 20% for all metrics in the test dataset. The RF declined by KS (9.09%), AUC (8.09%), F1-Score (16.47%), Recall (16.15%), PRAUC (16.46%), Precision (16.78%) and Accuracy (0.04%). Consequently, the RF model is selected as the final model for further use in real-time fraud detection and for Shap value explainability.

Table 2. Model performance on the test data set.

Model	KS	AUC	F1-Score	Recall	PRAUC	Precision	Accuracy
LR	0.9090	0.9517	0.0937	0.9308	0.4901	0.0493	0.9726
LDA	0.9090	0.9153	0.1408	0.8462	0.4616	0.0768	0.9843
KNN	0.9090	0.9337	0.5622	0.8692	0.6424	0.4154	0.9979
CART	0.9090	0.8949	0.4671	0.7923	0.5619	0.3312	0.9972
NB	0.9090	0.9261	0.0971	0.8769	0.4643	0.0514	0.9752
SVM	0.9090	0.9293	0.1258	0.8769	0.4724	0.0678	0.9815
RF	0.9090	0.9191	0.8352	0.8385	0.8354	0.8321	0.9995
XGB	0.9090	0.9412	0.5239	0.8846	0.6285	0.3722	0.9976
Light GBM	0.9090	0.9339	0.6141	0.8692	0.6721	0.4748	0.9983

4.3. Final Model Output and Adjustment

Following Visa's approach to its Provisioning Intelligence [20] the model output (predicted probabilities) generated by the Random Forest (RF) model were rounded to two decimal points and then multiplied by 100 to bring them within a range of 0 to 100 with one-point increments. In this scale, a score of 100 denotes the highest risk, while a score of 0 indicates the lowest risk.

In real-time fraud detection, the final model output is also converted to a score tie with a set of rules for use in decisioning. Based on the score and the rules, a transaction could be approved or sent for manual review pending the customer's authentication or declined outright. Subsequently, these scores and predicted probabilities were sorted and equally grouped into 10 bins. To assess the appropriateness of this scoring approach, a histogram comparison of predicted probabilities and the score associated with fraudulent transactions in the test dataset was generated.

As depicted in Figure 2, fraudulent transactions exhibit distinct distributions across the binned probabilities and scores. The predicted probabilities and the scores for fraudulent transactions display a right-skewed distribution, with approximately 92% of fraudulent transactions falling within the highest bins, specifically in the ranges [0.9 to 1] and [90 to 100] for predicted probabilities and scores, respectively. Thus the conversion of the predicted probabilities to scores did not alter the distribution of fraud transactions.

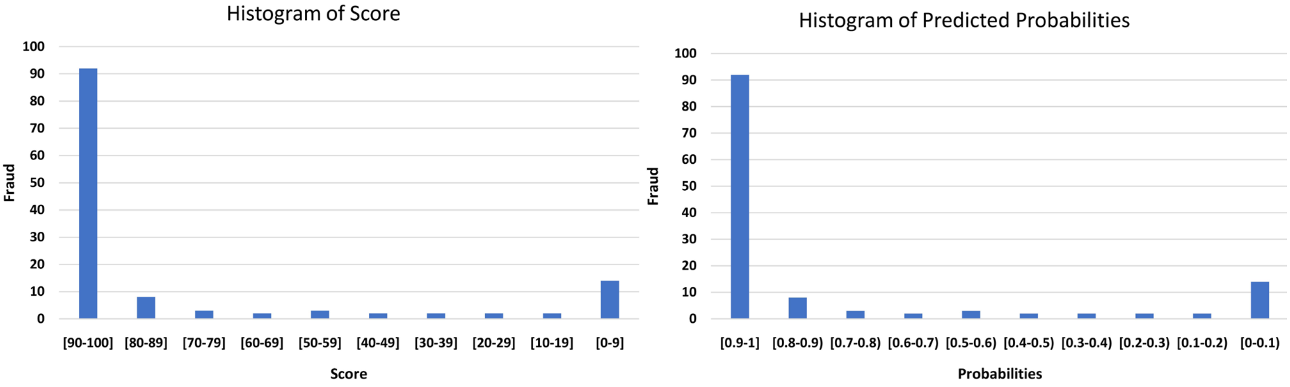

Figure 2. Histogram of predicted probabilities and scores for fraudulent transactions.

4.4. Detection Rate in the Test Data

To assess the detection rate, which represents the proportion of fraudulent transactions detected by the model, a total transaction count per score bin was tabulated, with each bin corresponding to a specific probability range. As illustrated in Table 3, the highest score bin, ranging from 90 to 100, aligns with the highest probability bin [0.9 - 1]. This particular bin contains 92 fraudulent transactions out of 100 transactions. In simpler terms, approximately 92% of transactions scoring 0.9 and above (riskiest bin) are identified as fraudulent, equating to a detection rate of approximately 70.77% at a lower false positive ratio of 0.09 for all fraudulent transactions in the test dataset.

Table 3. Fraud detection in the test data set.

Score	Probabilities	Non Fraud	Fraud	Total Transaction	Fraud Rate	Cum Non Fraud	Cum Fraud	Det Rate	FPR
90 - 100	[0.9 - 1]	8	92	100	92.00%	8	92	70.77%	0.09
80 - 89	[0.8 - 0.9)	1	8	9	88.89%	9	100	76.92%	0.09
70 - 79	[0.7 - 0.8)	1	3	4	75.00%	10	103	79.23%	0.1
60 - 69	[0.6 - 0.7)	5	2	7	28.57%	15	105	80.77%	0.14
50 - 59	[0.5 - 0.6)	8	3	11	27.27%	23	108	83.08%	0.21
40 - 49	[0.4 - 0.5)	6	2	8	25.00%	29	110	84.62%	0.26
30 - 39	[0.3 - 0.4)	16	2	18	11.11%	45	112	86.15%	0.4
20 - 29	[0.2 - 0.3)	42	2	44	4.55%	87	114	87.69%	0.76
10 - 19	[0.1 - 0.2)	217	2	219	0.91%	304	116	89.23%	2.62
0 - 9	[0 - 0.1)	85,009	14	85,023	0.02%	85,313	130	1	656.3
Total		85,313	130	85,443					

A similar trend is observed across the remaining score bins, where the fraud rate within each bin progressively decreases compared to the riskiest bin in terms of fraud risk. Table 3 also reveals that the RF model captures more than half of the fraud (resulting in a detection rate of over 50%) for each bin in the test dataset. These findings further affirm the Random Forest (RF) model's robust performance in detecting fraudulent activities in unseen data.

4.5. In Rank Ordering/Monotonicity Testing

The rank-ordering is a visual assessment that illustrates the model's ability to consistently rank the fraud rates in decreasing order as the scores decrease. To evaluate this ability, the scores are plotted against the corresponding fraud rates[3] in Table 3. As depicted in Figure 3, the plot exhibits an expected, monotonically decreasing trend in the marginal fraud rate across the score bins. This observation shows the model's robust rank-ordering ability, exemplified by the highest fraud rate of 92% recorded in the riskiest score bins [90 to 100]. This further reaffirms the best model (RF) model's ability in detecting fraudulent activities.

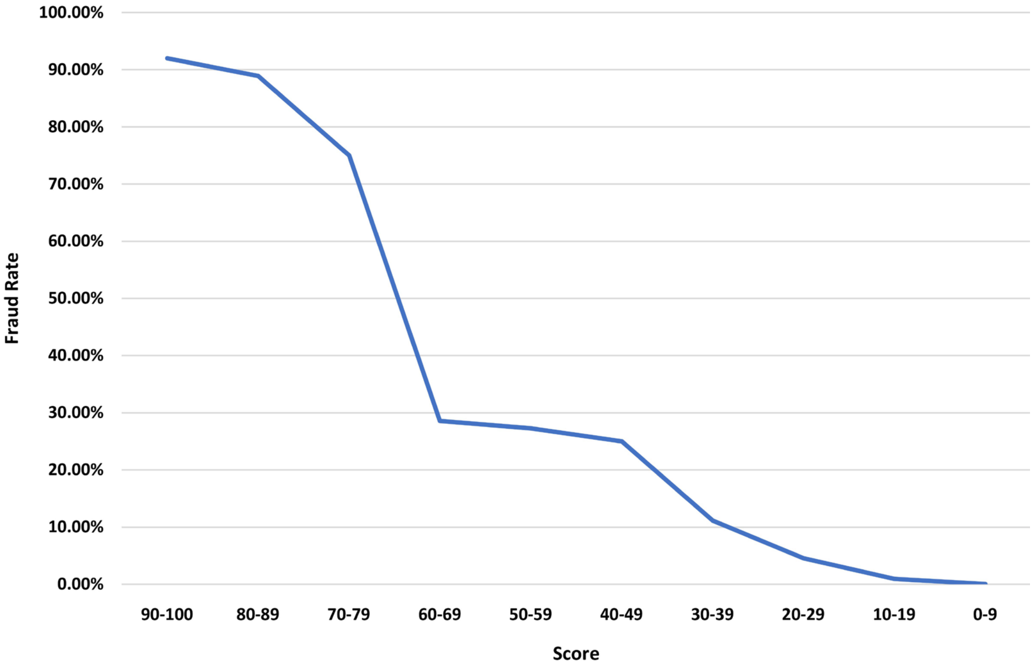

Figure 3. Rank ordering of fraud rates.

4.6. Distribution of Score by Fraud Tag

To conduct a visual examination of the model's output, a comparison of histograms depicting the distribution of scores derived from the predicted probabilities for the fraud labels (true outcomes) using the test data is generated. As depicted in Figure 4 on the left, the scores for legitimate transactions (fraud = 0) exhibit a right-skewed distribution. The peak of the distribution for legitimate transactions occurs at the lowest, least risky score (0), aligning with expectations and reflecting the low incidence of fraud rate in the test data.

[3] Fraud Rate is Fraud/Total Transaction for each bin.

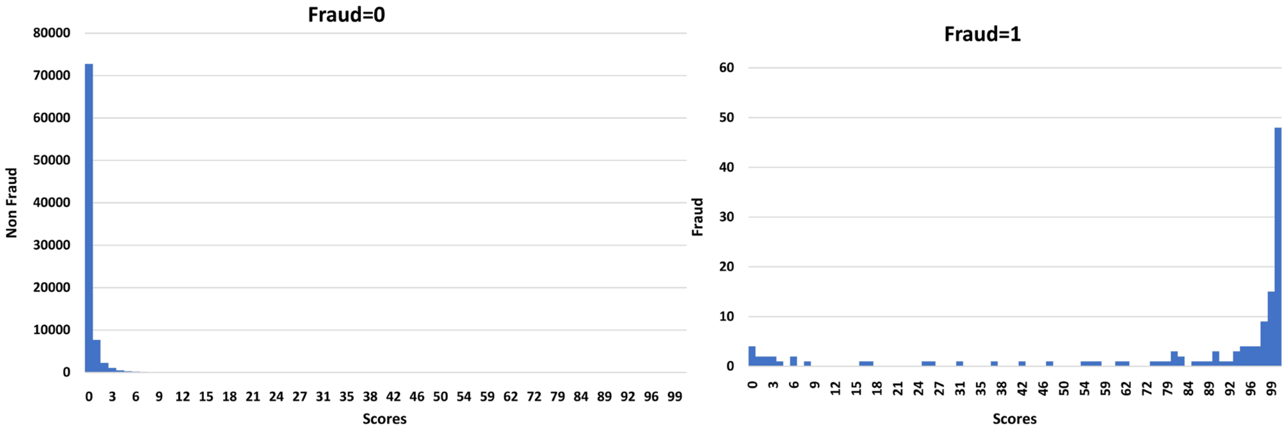

Figure 4. Histogram of score by fraud tag.

Conversely, the scores for fraudulent transactions (fraud = 1) display a left-skewed distribution, with a peak in the highest, riskiest score range, as expected. Based solely on visual inspection, the RF model's discriminatory power is reaffirmed as obtained by the KS and AUC as discussed in Table 1 and Table 2.

5. Explainability

Machine learning models are frequently characterized as "black boxes" because of their inherent complexity, which makes it challenging to understand the reasoning behind their predictions. Consequently, there is a growing demand for methods that render these models more explainable and interpretable, shedding light on the intricacies of their predictions [21] [22] [23]. ML explainability is important to ensure algorithmic fairness, identify potential bias in the training data, and ensure that the final model performs as expected [24].

Machine learning interpretability encompasses a wide array of techniques that are used to clarify and understand the decision-making processes of machine learning models. These techniques include feature importance scores, partial dependence plots, LIME (Local Interpretable Model-Agnostic Explanations), SHAP (Shapley Additive explanations) values, and many others. Among these, SHAP values are often preferred due to their robust theoretical foundation, consistency, and ability to explain complex models by providing coherent feature attributions.

5.1. Shapley Additive Explanations (SHAP)

The use of Shapley Additive explanations (SHAP) for explainability developed by Lundberg and Lee [25] is rooted in cooperative game theory, which distributes the total gain among players according to their respective contributions. Within the game theory framework, the model represents the game's rules, and the input features are hypothetical individuals who could either play the game (an observed feature) or not (an unobservable characteristic). Therefore, the SHAP technique determines the Shapley values by assessing the model under various feature combinations and figuring out the average difference in the prediction (outcome) between the presence and absence of a feature.

5.2. Global Explainability: SHAP Value Feature Importance Plot

I generated a SHAP summary plot to gain a deeper insight and create a more informative plot to visualize the feature's importance. This plot organizes the contributions of the features from the most influential to the least influential. Figure 5 summarizes each feature's impact on the final prediction for the test dataset. The plot clearly shows that "V12" is the most influential feature when making predictions, followed closely by "V14." Conversely, "V27" is identified as the least important feature, exerting minimal influence on the model's predictions.

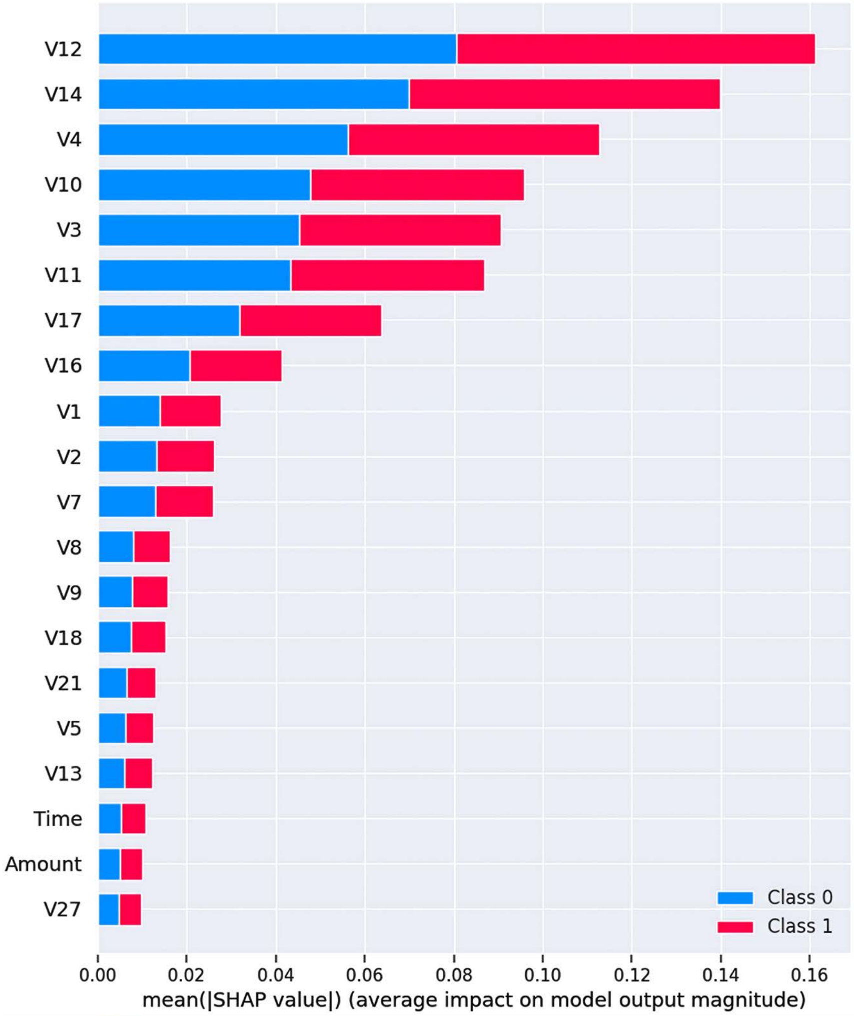

Figure 5. Shap value importance score.

5.3. Local Explainability

The SHAP values force plot provides local interpretability for each data point predicted by the model. In Figures 6-8, I present force plots generated using SHAP values for the 15th, 20th, and 40th instances, respectively.

These force plots vividly demonstrate how specific features, each with its unique contribution, influence the prediction for each of these instances. Some features exert a positive impact, pushing the prediction higher, while others have a negative effect, pulling it lower. The cumulative effect of all feature contributions adds up to the final prediction value.

In these force plots, features with red coloring indicate contributions that increase the model's prediction, while features with blue coloring indicate contributions that decrease the prediction. The intensity of the color reflects the magnitude of the contribution. Wider bars on the plot indicate a more extensive range of values for a feature, emphasizing its more significant influence on the prediction.

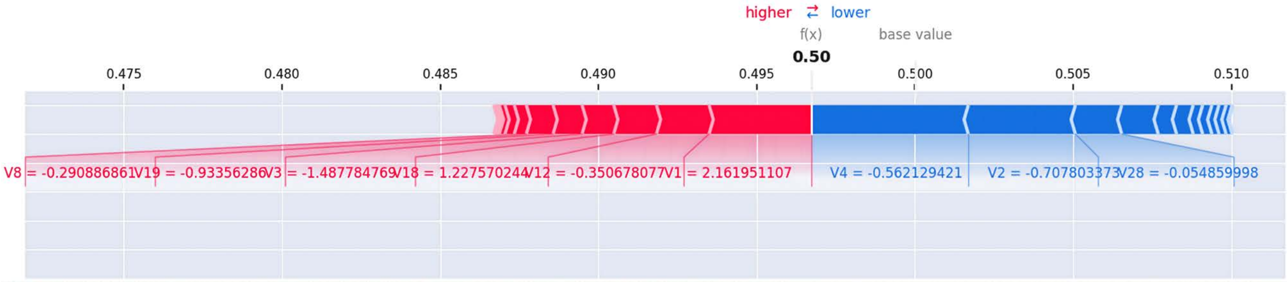

Figure 6. Shap value explanation for instance 15.

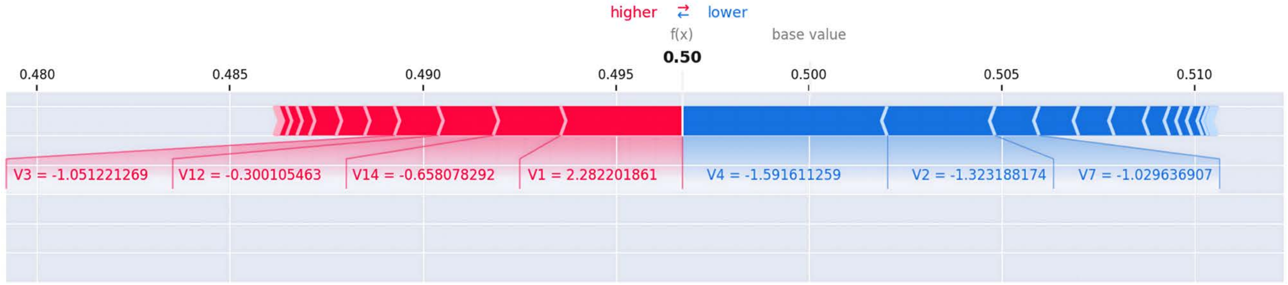

Figure 7. Shap value explanation for instance 20.

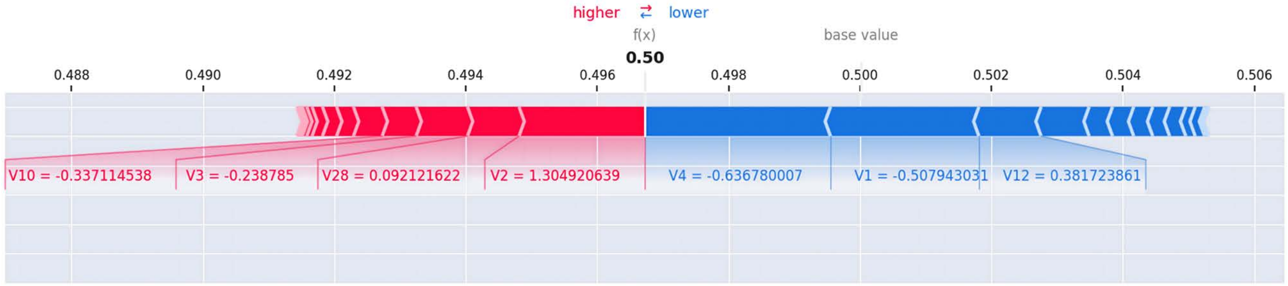

Figure 8. Shape value explanation for instance 40.

6. Conclusions

The study focused on fraud detection using machine learning; I employed a credit card dataset sourced from Kaggle. After preprocessing and feature selection, I evaluated several machine learning models, including Logistic Regression, Linear Discriminant Analysis, K-nearest Neighbors, Classification and Regression Tree, Naive Bayes, Support Vector Machine, Random Forest, XGBoost, and Light Gradient-Boosting Machine.

The Random Forest model emerged as the top-performing model, showcasing remarkable results in distinguishing between fraudulent and non-fraudulent transactions. It exhibited high accuracy, precision, recall, F1 score, and AUC, demonstrating its robustness in identifying fraudulent activities.

To assess overfitting, I evaluated model performance on a separate test dataset, and the Random Forest model maintained its strong performance, indicating its ability to generalize effectively.

I also explored model interpretability using SHAP (Shapley Additive exPlanations) values. The SHAP summary plot highlighted the importance of individual features, with "V12" being the most influential and "V14" closely following. Additionally, SHAP force plots provided local interpretability, revealing how specific features impacted predictions for individual instances.

In conclusion, the Random Forest model, supported by SHAP values for explainability, represents a powerful tool for real-time fraud detection in credit card transactions. Its strong performance and interpretability make it a valuable asset for financial institutions seeking to enhance security and minimize fraudulent activities.

7. Limitation and Future Work

While this study demonstrates the effectiveness of the Random Forest model for real-time credit card fraud detection, there are some limitations to consider. Firstly, although widely used in similar studies, the dataset used in this study is from a European credit card issuer and may not fully represent global fraud patterns. Future research could explore the model's performance on datasets from different geographical regions to assess its generalizability. Also, as discussed above, PCA transformation was applied to the original data, excluding identifiable information features such as "time" and "amount." Applying PCA transformation to original data for machine learning introduces challenges such as loss of interpretability and difficulty explaining results due to the transformation of features into orthogonal components. Additionally, PCA may lead to information loss, especially if essential information is discarded. Despite these challenges, PCA remains a valuable technique for dimensionality reduction; future research could explore methods to utilize actual features to mitigate the loss of interpretability.

Secondly, credit card fraud techniques continuously evolve, and fraudsters adapt their strategies to circumvent detection systems. Therefore, the model's performance may degrade if not regularly updated with new, representative data. Future work could investigate online shopping habits to enable the model to dynamically adapt to emerging fraud patterns.

The RF model is chosen as the best performing model because it requires few hyperparameters to optimize which is advantageous given the limited computational resources. Moreover, the performance of a model could vary depending on the specific characteristics of the data and the problem at hand. The RF model has performed better than other classification models on the European data in the literature. Thus, it would be essential for future research to experiment with another dataset with high computational resources to determine the performance of the model in a different environment or to actual credit card fraud detection system. I would also recommend for future work to consider the real-time and scalability of the model, as well as how it handles the challenges of large-scale data and high-speed transaction traffic.

Lastly, this study focuses on the binary classification of transactions as fraudulent or non-fraudulent. However, fraud detection systems often incorporate additional actions in real-world scenarios, such as manual review or authentication challenges. Future work could extend the model to a multi-class classification problem, incorporating these intermediate actions to better align with real-world fraud management strategies.

8. Policy Implication

The findings from this study have significant policy implications for financial institutions, regulators and consumers. The demonstrated effectiveness of machine learning, particularly the Random Forest Model, in detecting credit card fraud in real-time highlights the potential for these techniques to enhance security measures and protect consumers from financial losses.

Financial institutions should consider integrating machine learning-based fraud detection systems into their existing risk management frameworks. By leveraging the power of real-time fraud detection, banks and credit card issuers can proactively identify and prevent fraudulent transactions, reducing financial losses and minimizing the impact on affected customers.

Regulators and policymakers should encourage the adoption of advanced fraud detection technologies, such as machine learning, to strengthen the overall security of financial systems. This could involve providing guidance and incentives for final institutions to invest in these technologies and establishing standards for their implementation and monitoring.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Delamaire, L., Abdou, H. and Pointon, J. (2009) Credit Card Fraud and Detection Techniques: A Review. *Banks and Bank Systems*, **4**, 57-68.

[2] Capital One (2023) How to Spot and Avoid Credit Card Skimmers. https://www.capitalone.com/learn-grow/privacy-security/credit-card-skimmers/

[3] Barker, K.J., D'Amato, J. and Sheridon, P. (2008) Credit Card Fraud: Awareness and Prevention. *Journal of Financial Crime*, **15**, 398-410. https://doi.org/10.1108/13590790810907236

[4] Consumer Sentinel Network Data Book 2022. Federal Trade Commission. https://ftc.gov/

[5] New FTC Data Show Consumers Reported Losing Nearly $8.8 Billion to Scams in 2022. Federal Trade Commission.

[6] Federal Bureau of Investigation (2022) Federal Bureau of Investigation Internet Crime Report. https://www.ic3.gov/Media/PDF/AnnualReport/2022_IC3Report.pdf

[7] Thennakoon, A., Bhagyani, C., Premadasa, S., Mihiranga, S. and Kuruwitaarachchi, N. (2019) Real-Time Credit Card Fraud Detection Using Machine Learning. 2019 9*th International Conference on Cloud Computing, Data Science & Engineering* (*Confluence*), Noida, 10-11 January 2019, 488-493. https://ieeexplore.ieee.org/document/8776942 https://doi.org/10.1109/CONFLUENCE.2019.8776942

[8] Abakarim, Y., Lahby, M. and Attioui, A. (2018) An Efficient Real Time Model for Credit Card Fraud Detection Based On Deep Learning. *Proceedings of the* 12*th International Conference on Intelligent Systems*: *Theories and Applications*, Rabat, 24-25 October 2018, 1-7. https://doi.org/10.1145/3289402.3289530

[9] Yee, O.S., Sagadevan, S. and Ahamed, H. (2018) Credit Card Fraud Detection Using Machine Learning As Data Mining Technique. *Journal of Telecommunication*, *Electronic and Computer Engineering*, **10**, 23-27.

[10] Karunachandra, B., Putera, N. Wijaya, S.R., Suryani, D., Wesley, J. and Purnama, Y. (2023) On the Benefits of Machine Learning Classification in Cashback Fraud Detection. *Procedia Computer Science*, **216**, 364-369. https://doi.org/10.1016/j.procs.2022.12.147

[11] Varmedja, D., Karanovic, M., Sladojevic, S., Arsenovic, M. and Anderla, A. (2019) Credit Card Fraud Detection—Machine Learning methods. 2019 18*th International Symposium INFOTEH-JAHORINA* (*INFOTEH*), East Sarajevo, 20-22 March 2019, 1-5. https://doi.org/10.1109/infoteh.2019.8717766

[12] Sailusha, R., Gnaneswar, V., Ramesh, R. and Rao, G.R. (2020) Credit Card Fraud Detection Using Machine Learning. 2020 4*th International Conference on Intelligent Computing and Control Systems* (*ICICCS*), Madurai, 13-15 May 2020, 1264-1270. https://ieeexplore.ieee.org/abstract/document/9121114 https://doi.org/10.1109/ICICCS48265.2020.9121114

[13] Raghavan, P. and Gayar, N.E. (2019) Fraud Detection Using Machine Learning and Deep Learning. 2019 *International Conference on Computational Intelligence and Knowledge Economy* (*ICCIKE*), Dubai, 11-12 December 2019, 334-339. https://ieeexplore.ieee.org/document/9004231 https://doi.org/10.1109/ICCIKE47802.2019.9004231

[14] Dornadula, V.N. and Geetha, S. (2019) Credit Card Fraud Detection Using Machine Learning Algorithms. *Procedia Computer Science*, **165**, 631-641. https://doi.org/10.1016/j.procs.2020.01.057

[15] Awoyemi, J.O., Adetunmbi, A.O. and Oluwadare, S.A. (2017) Credit Card Fraud Detection Using Machine Learning Techniques: A Comparative Analysis. 2017 *International Conference on Computing Networking and Informatics* (*ICCNI*), Lagos, 29-31 October 2017, 1-9. https://doi.org/10.1109/iccni.2017.8123782

[16] Tran, P.H., Tran, K.P., Huong, T.T., Heuchenne, C., HienTran, P. and Le, T.M.H.

(2018) Real Time Data-Driven Approaches for Credit Card Fraud Detection. *Proceedings of the* 2018 *International Conference on E-Business and Applications*, New York, 23-25 February 2018, 6-9. https://doi.org/10.1145/3194188.3194196

[17] Batani, J. (2017) An Adaptive and Real-Time Fraud Detection Algorithm in Online Transactions. *International Journal of Computer Science and Business Informatics*, **17**, 1-12.

[18] Khan, A.U.S., Akhtar, N. and Qureshi, M.N. (2014) Real-Time Credit-Card Fraud Detection Using Artificial Neural Network Tuned by Simulated Annealing Algorithm. *Proceedings of International Conference on Recent trends in Information, Telecommunication and Computing*, *ITC*, Chandigarh, 113-121.

[19] Kaggle (2019) Credit Card Fraud Detection. https://www.kaggle.com/

[20] Visa (2023) Visa Provisioning Intelligence Launches to Combat Token Fraud. https://investor.visa.com/news/news-details/2023/Visa-Provisioning-Intelligence-Launches-to-Combat-Token-Fraud/default.aspx

[21] Shao, Y., Cheng, Y., Shah, R.U., Weir, C.R., Bray, B.E. and Zeng-Treitler, Q. (2021) Shedding Light on the Black Box: Explaining Deep Neural Network Prediction of Clinical Outcomes. *Journal of Medical Systems*, **45**, Article 5. https://doi.org/10.1007/s10916-020-01701-8

[22] Linardatos, P., Papastefanopoulos, V. and Kotsiantis, S. (2020) Explainable AI: A Review of Machine Learning Interpretability Methods. *Entropy*, **23**, Article 18. https://doi.org/10.3390/e23010018

[23] Roscher, R., Bohn, B., Duarte, M.F. and Garcke, J. (2020) Explainable Machine Learning for Scientific Insights and Discoveries. *IEEE Access*, **8**, 42200-42216. https://doi.org/10.1109/access.2020.2976199

[24] Gilpin, L.H., Bau, D., Yuan, B.Z., Bajwa, A., Specter, M. and Kagal, L. (2018) Explaining Explanations: An Overview of Interpretability of Machine Learning. 2018 *IEEE* 5*th International Conference on Data Science and Advanced Analytics* (*DSAA*), Turin, 1-3 October 2018, 80-89. https://doi.org/10.1109/dsaa.2018.00018

[25] Lundberg, S.M. and Lee, S.-I. (2017) A Unified Approach to Interpreting Model Predictions. *Proceedings of the* 31*st International Conference on Neural Information Processing Systems*, New York, 4-9 December 2017, 4768-4777. https://neurips.cc/