# Framework for Credit Card Fraud Detection Using Benefit-Based Learning and Periodic Features

**Shellyann Sooklal** ( shellyann.sooklal@gmail.com )
University of the West Indies
**Patrick Hosein**
University of the West Indies

**Research Article**

**Keywords:** Credit-card fraud, Cost-sensitive learning, Benefit-based analytics, Imbalance data, Binary classification, Periodic features

**Posted Date:** March 14th, 2023

**DOI:** https://doi.org/10.21203/rs.3.rs-2652853/v1



**Additional Declarations:** No competing interests reported.

# Framework for Credit Card Fraud Detection Using Benefit-Based Learning and Periodic Features

Shellyann Sooklal[1] and Patrick Hosein[1]

[1]Department of Computer Science, The University of the West Indies, St. Augustine, Trinidad.

Contributing authors: shellyann.sooklal@gmail.com; patrick.hosein@sta.uwi.edu;

**Abstract**

Online credit card fraud is an ongoing problem and with the recent COVID-19 pandemic, there has been a surge of merchants moving their businesses online. It is therefore crucial to identify fraudulent activities before it causes loss to both the bank and its customers. Due to the dynamic nature of fraudsters as well as customer spending behavior, machine learning algorithms are appropriate for this task. However, credit card fraud data is typically imbalanced, favoring the positive class (legitimate transactions), causing traditional machine learning algorithms to err on the side of this majority class; since they consider equal costs and benefits for different decision outcomes when training. Nevertheless, it is more beneficial to correctly identify fraudulent transactions. Therefore, in this paper, we propose a technique for identifying credit card fraud that first accounts for customer spending patterns by aggregating transactions to creative new features based on periodic data. Then, we consider benefits and costs when training an XGBoost classifier in order to achieve maximum benefits. We also evaluate the performance of the classifier using benefits and costs. We demonstrate the effectiveness of our approach using data provided by a bank.

**Keywords:** Credit-card fraud, Cost-sensitive learning, Benefit-based analytics, Imbalance data, Binary classification, Periodic features

## 1 Introduction

Over the past few decades there has been significant growth in e-commerce activities and with the recent COVID-19 pandemic, these activities have increased exponentially. Also, more merchants are moving their businesses online. This escalates the risk and frequency of fraudulent e-commerce transactions which can result in significant financial loss both to banks and their customers. Detection of this type of fraud needs to be immediate since upfront detection can prevent recovery efforts and hence save the bank and its customers from eventual financial loss especially since customers may not be aware of fraudulent activities on their account as soon as it occurs. In addition, the detection process needs to be efficient due to the large number of transactions occurring in real-time.

Furthermore, online fraudulent transactions are continuously evolving and are disperse over multiple customer accounts therefore it is increasingly difficult for current rule-based systems to detect and prevent. According to the 2021 Nilson Report [20], the global loss due to credit card fraud in 2021 amounted to $32.20 billion which was a 14% increase over the total loses in 2020 ($28.58 billion). The report also projected the total loss

due to fraud to increase to $49.32 billion by 2030. The work presented in this paper was performed in collaboration with a bank, where for the period November 2018 to July 2020 there were approximately 5,000 instances of online fraud, most of which were not detected by the current rule-based system and hence resulted in significant loss to the bank. Hence, due to the nature of the problem, machine learning algorithms are quite suited for this task especially since they have the ability to continuously learn and evolve with new data and can also be quite efficient. Many researchers have already applied machine learning techniques in an attempt to improve e-commerce fraud detection [16, 17, 19] and their approaches have shown to improve the detection rate of online fraudulent activities while still maintaining low false positives. The algorithms also show improved results given the diversity of fraud datasets. The choice of algorithm depends on the type of fraud being detected as well as the properties of the data available. Hence, the main objective of the research is to apply appropriate machine learning techniques to the datasets provided by the bank in order to develop a suitable model for detection of e-commerce fraud transactions in real time.

In order to achieve the objectives of this research we need to take into account the properties of the data. First, customers typically have specific patterns when making purchases online. For example, they may usually make purchases from specific locations and devices and may have specific merchants who they purchase from. They may also have a spending range. All of these patterns need to be considered when detecting fraud since any deviation from a customer's spending behaviour can be an indicator of fraud.

Secondly, credit card fraud datasets are typically skewed towards the positive class (legitimate transactions). That is, the dataset generally contains a large number of legitimate cases and a very small number of fraudulent cases. This imbalance of the dataset causes machine learning algorithms to favor the side of the legitimate class. This is due to the fact that machine learning algorithms consider equal costs for misclassification of the two classes. This is also true for commonly used performance metrics such as accuracy and area under the curve. However, for the case of credit card fraud detection, this is not ideal. It is more critical to correctly identify a fraudulent case than a legitimate case, since the costs associated with misclassifying a fraudulent case is much higher than the cost associated with misclassifying a legitimate case. This is also true when considering the benefits due to correct classification. The benefits of correctly classifying a fraudulent case far outweighs the benefits of correctly identifying a legitimate case. Therefore, for this research we consider including costs and benefits when training the machine learning algorithms as well as evaluating their performance.

The goal of the bank is to significantly improve upon the detection rate of the current rule-based system; hence reducing financial loss to both the bank and its customers. In addition, the project aims at creating a model to improve the accuracy of the bank's offline detection system. For both the offline and online models, the bank would like to achieve a false negative (fraudulent transactions that were not detected) to true positive (fraudulent transactions which were correctly identified) ratio of 5:1 or lower. The bank supplied the required datasets for training and testing of the machine learning algorithms. The datasets comprise of past transactions with instances of both fraudulent cases and legitimate cases. Since the labels from the current fraud detection system are not 100% accurate, a list of fraudulent transactions was also provided so that the instances in the dataset can be labeled.

In summary, the main contributions of this paper is a credit card fraud detection algorithm which combines the use of historical patterns of customer purchase history with a benefit-based classification algorithm (specifically the XGBoost classifier), which significantly improves the detection rate of the bank's current rule-based system. The rest of the paper is as follows. Section 2 summaries related work in the field, Section 3 provides a detailed explanation of the dataset and machine learning pipeline used to achieve the goals of the research project, Section 4 presents and discusses the results of the experiments and Section 5 concludes the main findings.

## 2 Related Work

As mentioned earlier, due to machine learning algorithms' efficiency and ability to continuously learn patterns from past transactions in order to detect fraud, many researchers have already

jumped on this opportunity and have applied multiple machine learning algorithms and approaches to improve credit card fraud detection.

Some researches have either applied a single algorithm or took the approach to compare multiple algorithms to determine which one gave the best performance with the fraudulent cases. Some of the main machine learning algorithms used by these researchers include

- Logistic Regression [8, 10–12, 22, 28, 29]
- Support Vector Machines [21, 22, 27, 29, 30]
- Neural Networks [1, 3, 5, 13, 18, 26, 27, 31]
- Decision Trees [6, 7, 11, 18, 22, 28]
- Random Forests [6–8, 11, 12, 15, 21, 22, 28]
- Naive Bayes [11, 27–29]
- K-Nearest Neighbors [11, 22, 29]
- Isolation Forest [13, 22, 23]
- Local Outlier Factor [10, 13, 23]

Random Forests and Neural Networks generally produced good results, however, some researchers reported that Neural Networks took a long time to train. In general, the best suited algorithm depends on the properties of the data at hand, hence, the reason for many researchers taking the route of comparing the algorithms to determine the one that was most appropriate.

Customers tend to have spending behaviours and these behaviours change over time. Due to this, some researchers have included the detection of concept drifts as part of their solution. [9] handled concept drift both as an active solution by identifying changes in statistics and a passive solution by continuously updating the model using new records. [14] used face to face transactions in order to compute the distance in concept drift between consecutive transactions and added this as a new feature to the model. [24] handled concept drift via a transaction window bagging approach. On the other hand, instead of using concept drift, [32] analyzed periodic customer behaviour in order to creature new features. Because of the success of their approach [4] and [33] adapted [32]'s methodology into their work, and hence we have decided to apply this strategy to our work.

Due to the imbalance nature of credit card fraud datasets, researchers have employed strategies such as SMOTE to combat the imbalance of the data [2]. However, for this work we instead adapt an XGBoost classifier to account for benefits and costs. The XGBoost classifier uses the "binary:logistic" objective function, therefore we replace this function with a benefit-based logistic regression cost function that we proposed in [25]. Hence, in this paper we combine a transaction aggregation strategy using historical data as well as a benefit-based XGBoost classifier to achieve an improved fraud detection rate.

# 3 Methodology

For this research, we collaborated with a bank, whose details we cannot disclose. The goal of the project is to significantly improve upon the detection rate of the current rule-based system; hence reducing financial loss to the bank and its customers. In addition, the project aims at creating a model to improve the accuracy of the bank's offline detection system. For both the offline and online models, the bank would like to achieve a false negative (fraudulent transactions that were not detected) to true positive (fraudulent transactions which were correctly identified) ratio of 5:1 or lower.

## 3.1 Dataset Description

The bank supplied the required datasets for training and testing of the machine learning algorithms. The datasets comprise of past transactions with instances of both fraudulent cases and legitimate cases. Since the labels from the current fraud detection system are not 100% accurate, a list of fraudulent transactions was also provided so that the instances in the dataset can be labeled.

There are 3 datasets which were provided by the bank, let's call them T1, T2 and T3:

- T1 is the main transaction dataset and contains transaction data for the period April 2018-July 2020. The fields in the dataset comprise of an encoded card number, details of the transaction (such as amount, country, merchant), details of the device used as well as a risk advice of whether to deny or allow a transaction. The purpose of this dataset is for training and testing of the proposed model. In order to sync with the time period in the T2 file, only transactions from May 2018 onwards are considered.
- T2 is a list of know fraudulent transactions that were recorded in the time period November 2018

to July 2020. Since fraudulent transactions are recorded months after the transaction date, a later start date is noted for this file. The T2 dataset contains similar fields as the T1 file, as well as an authorization code and a code to describe the type of transactions. Transactions with an code of 5 or 6 are online transactions and hence these are the transactions of interest from the dataset. The purpose of the T2 file is to match fraudulent transactions to the transactions in the T1 file, in order to label the transactions in the T1 file as fraudulent or legitimate.

- Due to inconsistencies between the data in the T1 and the T2 files, a third dataset, T3, is used to match the transactions between the first two files. That is, this is a bridge dataset. Therefore, T3 contains matching transactions to the online transactions in the T2 dataset. The card number and authorization code fields from the T2 and T3 datasets are used to match transactions between these two datasets. The time of transaction as well as card number are used to match the T3 transactions to the T1 dataset.

## 3.2 Pipeline

### 3.2.1 Labeling of Transactions

The first step in the pipeline is labeling the transactions in the T1 dataset as fraudulent or legitimate. The T2 file contains the list of fraudulent transactions. Relevant transactions containing a transaction code of 5 or 6 were extracted from the T2 dataset and were then matched to the transactions in the T3 file based on card number and authorization code. The resulting merged dataset contained instances of known online fraudulent transactions with attributes from both the T2 and T3 datasets.

The merged dataset was then used to label the T1 transactions. The transaction time from the T3 fields as well as card number were used to search for matching rows of data from the T1 file. Therefore, the datasets were cleaned with respect to these fields before matching the transactions. There was a notable time difference of 4 hours between the transactions in the T1 dataset and those in the T3 dataset during daylight savings time and 5 hours otherwise. Also, since fraudsters sometimes performed the same transaction multiple times, within seconds or sometimes minutes of each other, a time difference of 2 minutes and 30 seconds was used to find the closest matches. All transactions from the T1 file that fell within this time difference were then searched for the closest match. Once found, the matching transaction was labeled as fraudulent. All labels were recorded in a new field "isFraud".

The T1 dataset contained an attribute "Fraud Status" which comprised of manual fraud labels from the bank. The "isFraud" attribute was updated to represent a fraudulent transaction for transactions which contained a fraud status of "Confirmed Fraud" or "Assumed Fraud".

### 3.2.2 ELT Operations

After the fraudulent and legitimate transactions were appropriately labeled, extract, load and transform (ELT) operations were performed on the data. All unnecessary/irrelevant fields were removed and extensive imputation operations were performed on columns with missing data. For example, conversion rates were calculated using the "Currency", "Amount" and "Amount (USD)" fields from populated instances and then these rates were applied to the "Amount" field to fill in missing values for "Amount (USD)". For attributes where missing data could not be accurately imputed, the missing values were filled with a value to represent this. There were some columns that were only filled if further investigation of the transaction was required. Since these fields may be helpful to the machine learning algorithm for detecting fraud, they were left as part of the dataset in order to investigate their contributions to the model. The missing values from these fields were also filled with an appropriate value.

After successful cleaning and data imputation, the non-numeric attributes were encoded into numeric representations.

### 3.2.3 Selection of Machine Learning Algorithm

As previously mentioned, past researchers have applied machine learning (ML) algorithms for credit card fraud detection and have been successful in improving the detection rate. In order to find a suitable ML algorithm for our purposes, multiple machine learning algorithms were applied to

the cleaned and encoded dataset and their performances were compared. The dataset was first split into training and test sets (75% of the data was used for training and the remaining 25% was used for testing). The training data was used to train each of the ML algorithms and the test set was used to evaluate and compare their results. The machine learning algorithms that were compared were:

- Logistic Regression
- K Nearest Neighbors
- Gaussian Naive Bayes
- Decision Tree
- Random Forest
- Support Vector Machine
- XGBoost

The algorithms were compared based on sensitivity and specificity. Sensitivity, also known as the true positive rate (TPR), is the ability of the classifier to correctly identify fraudulent cases. On the other hand, specificity, also known as true negative rate (TNR), is the ability of the classifier to correctly identify negative cases, that is, instances of legitimate transactions. These metrics are calculated as follows:

$$Sensitivity = TPR = \frac{TP}{TP + FN} \tag{1}$$

$$Specificity = TNR = \frac{TN}{TN + FP} \tag{2}$$

where TP, true positives, is the number of correctly identified fraudulent cases, FN, false negative, is the number of incorrectly classified fraudulent cases, TN, true negatives, is the number of correctly classified legitimate transactions and FP, false positives, is the number of incorrectly classified legitimate transactions.

From the results obtained, the XGBoost classifier performed the best in terms of sensitivity and specificity and also performance time; therefore, it was selected as the most appropriate model for the project.

### 3.2.4 Feature Engineering

The raw transaction data for a particular instance in the dataset does not give any insight into customer spending patterns which can be useful in identifying fraudulent transactions. Customers tend to purchase from specific merchants, use specific devices to make their purchases and can also have a specific time frame in which they make their purchases. All of these spending patterns can be learnt from historical data and any deviation from customer spending behaviour can be an indicator for potential fraudulent activity. [32] proposed a transaction aggregation strategy to handle the inclusion of historical insights as part of the transaction features. The approach has shown to provide great improvement in detection rate and thus have been employed by many researchers including [4] and [33]. Hence, we decided to include their approach as part of our solution.

In order to compute the aggregated features based on historical transactions, a time frame for selection of historical transactions need to be selected. It is essential to note that customers tend to change their spending patterns over time therefore care must be taken in choosing an appropriate time frame. For our analysis, we tested using aggregated features from different time frames (5 days, 10 days, 30 days and 180 days) in order to compare their effects on the XGBoost model.

In order to explain how the aggregated features were calculated, we will consider the "Browser" attribute. If the dataset contains the browsers "Google Chrome", "Internet Explorer" and "Safari", then 6 new features would be added to the dataset. These features are:

- GoogleChromeCount
- GoogleChromeAmt
- InternetExplorerCount
- InternetExplorerAmt
- SafariCount
- SafariAmt

For each transaction in the dataset, these fields will be populated with the number of times each of these browsers were used in the selected historical time period by the current customer (identified by their card number) and also the total amount of the purchases made with these browsers in the time frame. Therefore, if historical data shows a card number being highly associated with "Google Chrome" but never with "Safari" then there would be cause to flag a transaction if the browser in the current transaction is "Safari".

In cases where there is a large number of unique values for a particular attribute, for example "Merchant", then these values are grouped based on the number of times they were associated with fraudulent transactions. The aggregated features would then record data based on groups rather than individual values.

### 3.2.5 Benefit-Based XGBoost Classifier

The problem at hand, that is credit card fraud detection, is a binary classification problem. For binary classification, the XGBoost classifier uses the "binary:logistic" objective function which is the same as that of logistic regression. Therefore the goal of this objective function is to determine the parameters $\vec{\theta}$ by minimizing the cost function:

$$L(\vec{\theta}) \equiv \frac{1}{N}\sum_{i=1}^{N} L_i(\vec{\theta}) \tag{3}$$

where

$$L_i(\vec{\theta}) = -y_i \log(h_\theta(\vec{x}_i)) - (1-y_i)\log(1-h_\theta(\vec{x}_i)), \tag{4}$$

$N$ is the number of samples and $\vec{x}_i$ is a given feature vector. We can determine the posterior probability of $\vec{x}_i$ belonging to the positive class by

$$p_i = P(y=1|\vec{x}_i) = h_\theta(\vec{x}_i) = g(\vec{\theta}^T\vec{x}_i) \tag{5}$$

where $g(\cdot)$ represents the logistic sigmoid function denoted by

$$g(z) = \frac{1}{1+e^{-z}}. \tag{6}$$

and $h_\theta(\vec{x}_i)$ denotes the classification of $\vec{x}_i$ using the parameter vector $\vec{\theta}$.

This loss function considers equal costs for different type of errors (false negatives, false positives), therefore we need to include benefits and costs into this function. In [25] we introduced a benefit-based Logistic Regression where we altered the loss function to include benefits and costs. This new cost function focuses on maximizing benefits $b_{ij}$, where $b_{ij}$ represents the benefits achieved from classifying an instance/transaction of class $i$ as class $j$, instead of minimizing costs. Note that if $i = j$ then $b$ is positive ($> 0$) since we have correct classification, otherwise $b$ is negative ($\leq 0$) for incorrect classification. The new cost function is defined as

$$L^B(\vec{\theta}) \equiv \frac{1}{N}\sum_{i=1}^{N} L_i^B(\vec{\theta}) \tag{7}$$

where

$$\begin{aligned} L_i^B(\vec{\theta}) = \; & y_i[h_\theta(\vec{x}_i)b_{11} + (1-h_\theta(\vec{x}_i))b_{10}] + \\ & (1-y_i)[h_\theta(\vec{x}_i)b_{01} + (1-h_\theta(\vec{x}_i))b_{00}]. \end{aligned} \tag{8}$$

This can be rewritten as

$$\begin{aligned} L_i^B(\vec{\theta}) = \; & y_i h_\theta(\vec{x}_i)(b_{11}-b_{10}) + b_{10} \\ & (1-y_i)(1-h_\theta(\vec{x}_i))(b_{00}-b_{01}) + b_{01} \end{aligned} \tag{9}$$

The function will be maximized with respect to $\vec{\theta}$. Therefore, we can safely remove $b_{10}$ and $b_{01}$ from the function, since they are constants, without having any influence on the optimal $\vec{\theta}$. Also, we can multiply the function by -1 in order to convert the problem to a minimization problem. Furthermore, we can take the resulting function and divide it by $(b_{11} - b_{10})$ without affecting the optimal solution. With these changes we get

$$L_i^B = -y_i h_\theta(\vec{x}_i) - (1-y_i)(1-h_\theta(\vec{x}_i))\eta \tag{10}$$

where $\eta$ is denoted by

$$\eta \equiv \frac{b_{00}-b_{01}}{b_{11}-b_{10}}. \tag{11}$$

This is similar to the Logistic Regression cost function but here we scale the error for class 0 by $\eta$. Therefore, in [25] we proposed using the logistic cost function along with this scaling in order to accommodate benefits while training. Therefore, the cost function can be written as

$$L_i(\vec{\theta}) = -y_i \log(h_\theta(\vec{x}_i)) - \eta(1-y_i)\log(1-h_\theta(\vec{x}_i)). \tag{12}$$

For the XGBoost classifier we also need to consider the first and second order derivatives which updates the loss function based on what was predicted by the model as well as the actual labels for the samples. These are defined as

$$gradient = h_\theta(\vec{x}_i) - y_i \tag{13}$$

$$hessian = h_\theta(\vec{x}_i)(1-h_\theta(\vec{x}_i)) \tag{14}$$

Following from Equation 12, we can include benefits using $\eta$ as follows:

$$gradient = h_\theta(\vec{x}_i)(\eta - \eta y_i + y_i) - y_i \quad (15)$$

$$hessian = (y_i + \eta - \eta y_i)h_\theta(\vec{x}_i)(1 - h_\theta(\vec{x}_i)) \quad (16)$$

**Benefit Model Based on Financial Loss**

We have defined a new loss function and first and second derivatives which include benefits and costs. We now need to determine values for benefits (true positives, true negatives) and costs (false positives, false negatives). We will show an example using financial loss due to fraud.

Due to confidentiality reasons, we cannot use actual values for financial loss due to fraud. Therefore, instead, we will use a hypothetical example in order to derive benefit and cost values. Let us first consider a baseline case where a transaction is non-fraudulent and nothing was done in terms of determining if the transaction was fraudulent or not. There would be no loss to the bank in terms of cost of investigating the case. Also, there would be no loss to the customer since the transaction is legitimate. This would equate to the case of $b_{00}$ and since there is no financial loss or savings for this scenario, we can set $b_{00} = 0$. On the other hand, if no check is done for a transaction that is fraudulent then there would be a financial loss to the customer since the transaction would go by undetected. Let's assume the average amount charged to a credit card to perform a fraudulent transaction is 700 USD. Then, we can set $b_{10} = -700$.

Let us now consider the scenario where checks are performed to determine if a transaction is fraudulent. If a fraudulent transaction is detected then this would be a savings to not only the customer but also to the bank in preventing overhead costs. If we average overhead costs to be around 200 USD, then the total savings would be 700 USD $(700 + 200)$. Therefore, we can let $b_{11} = 900$. However, if we misclassify a non-fraudulent transaction as fraudulent then there would be overhead costs to the bank in trying to resolve the case. Again, if these overhead costs are on average 200 USD then we can set $b_{01} = -200$. Using these benefits and costs values, we get $\eta = 0.125$.

### 3.2.6 Evaluation

Machine Learning algorithms were evaluated and compared based on sensitivity and specificity. From these evaluations, the XGBoost model was selected and further evaluated based on False Negative (FN) to True Positive (TP) ratio. The aim is to achieve a ratio of 5:1 or lower.

The results of the XGBoost model were compared to the results achieved when using the aggregated features, and also when using both aggregated features and the benefit-based XGBoost classifier.

In summary, the following models were compared on sensitivity and FN:TP ratio:

1. The bank's current rule-based system
2. XGBoost with raw transaction features
3. XGBoost with both raw and aggregated features computed from:
   (a) 5 days historical data
   (b) 10 days historical data
   (c) 30 days historical data
   (d) 180 days historical data
4. Benefit-based XGBoost (with respect to the benefit and costs values depicted in the financial loss scenario) using both raw and aggregated features computed using historical data that were based on the number of days which produced the best results from option 3 above

A maximum of 180 days was used to compute aggregated data since customer spending behaviour changes with time. The model from the above list which gave the best sensitivity value as well as the lowest FN:TP ratio was selected to be used in the offline fraud detection system.

**Benefit-Based Performance Metric**

In addition to comparing the models via sensitivity and FN:TP ratios, and since we are using a benefit-based XGBoost model as one of the options listed above, we need to determine if this model is achieving improved benefits compared to the accuracy based XGBoost model. In [25], we introduced a benefit based performance metric for binary classification. The main idea is as follows. A classifier produces a continuous score $s(\vec{x})$ for a given sample $\vec{x}$ in order to determine which class it belongs to. Assuming that scores for instances of class 0 are typically lower than scores for class 1, we can decide on a threshold $t$

where samples with a score less than or equal to $t$, that is $s(\vec{x}) \leq t$, then the sample is classified as belonging to class 0. Otherwise, if $s(\vec{x}) > t$, then the sample is classified as belonging to class 1. For our XGBoost classifier, we can let $f_0(s)$ represent the probability density function for scores belonging to class 0 and $f_1(s)$ represent the probability density function for scores belonging to class 1. The matching cumulative distribution functions can then be defined as $F_0(s)$ and $F_1(s)$. Following from the definition of $b_{ij}$ presented in Section 3.2.5, we can denote the prior probability of class $j \in \{0, 1\}$ by $\pi_j$.

If we let the $\pi_0 F_0(t)N$, that is the product of the probability of a sample belonging to class 0 times the probability of correct classification times the score threshold $t$, represent the number of samples from class 0 that are expected to be classified correctly given the threshold $t$, then we can define the expected benefit as

$$B(t) = \pi_0 F_0(t) b_{00} + \pi_0(1 - F_0(t)) b_{01} + \pi_1 F_1(t) b_{10} + \pi_1(1 - F_1(t)) b_{11}. \quad (17)$$

$b_{00}$ and $b_{11}$ are the only two positive benefit values, therefore we can maximize the expected benefit when $F_0(t) = 1$ and $F_1(t) = 0$ which can only occur when there is no overlap between the two distributions. We can therefore have an upper bound defined by $\pi_0 b_{00} + \pi_1 b_{11}$. If we let $B_\gamma$ denote the expected benefit for a classifier $\gamma$, then the performance metric can be defined as

$$\mu_\gamma \equiv \frac{B_\gamma}{\pi_0 b_{00} + \pi_1 b_{11}}. \quad (18)$$

One should note that the if $\mu_\gamma \approx 1$, then the classifier is performing close to optimal. Since, in general, we are maximizing $B(t)$, we can achieve optimality by finding the derivative of $B(t)$ with respect to $t$ and then by setting the result of this to zero. We would then end up with

$$f_0(t^*)\pi_0(b_{00} - b_{01}) = f_1(t^*)\pi_1(b_{11} - b_{10}). \quad (19)$$

### 3.2.7 Mapping Decision Tree to Legacy Rules

The current real-time system uses set rules in order to determine if a transaction is fraudulent or legitimate. The option to reference a script to run additional rules is not available at this time therefore the best tree from the benefit-based XGBoost model was extracted in order to map the model rules to the current system.

## 4 Discussion of Results

After performing data labeling and cleaning, the dataset comprised of approximately 1,070,000 legitimate transactions and 5,000 fraudulent transactions.

Figure 1 illustrates the comparison of the different machine learning algorithms based on sensitivity and specificity, after they were applied to the dataset. Both the Logistic Regression and the Support Vector Machine algorithms favoured the negative class, that is, legitimate transactions. These models classed all transactions as legitimate and hence had a specificity value of 100% since all actual legitimate transactions would be correctly classified. However, all fraudulent transactions were incorrectly classified therefore the sensitivity value was 0%. On the other hand, Guassian Naive Bayes favored the fraudulent class. The algorithm classed most of the instances as fraudulent and hence had a high sensitivity value (97.47%) but low specificity value (10.88%). Since the dataset comprises of mainly legitimate instances the overall accuracy of this model was low (11.29%).

The remaining algorithms all generally performed well; however, the XGBoost classifier gave the best performance in terms of sensitivity, specificity and speed (training time). Therefore, the XGBoost classifier was selected for the project. The feature importance graph from the XGBoost model is shown in Figure 2.

From the feature importance graph, the "Transaction ID" attribute is one of the main contributors. On further investigation into the fraudulent transactions, it was discovered that fraudsters sometimes perform the same transaction multiple times, in a short space of time. The difference in time between these transactions range from a few seconds to a few minutes. This results in somewhat consecutive "Transaction ID" values for a particular card number and hence can be a factor in determining if a transaction is fraudulent. This scenario is illustrated in Table 1.

"Card Number" is the top contributor in the feature importance graph. The reason for this is that "Card Number" needs to be paired with other

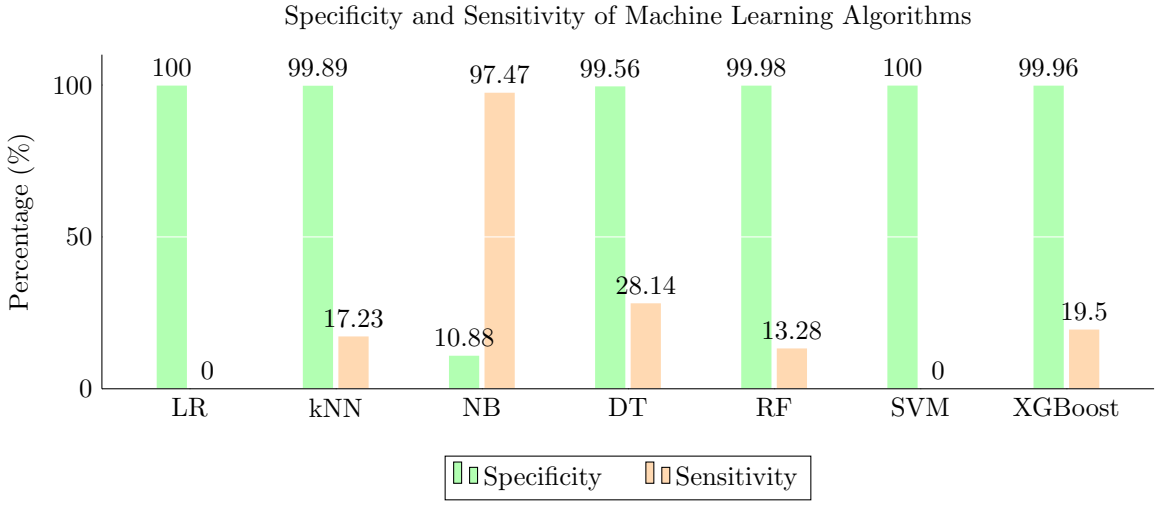


**Fig. 1** Comparison of Machine Learning Algorithms (*LG = Logistic Regression, kNN = k Nearest Neighbors, NB = Guassian Naive Bayes, DT = Decision Tree, RF = Random Forest, SVM = Support Vector Machine)

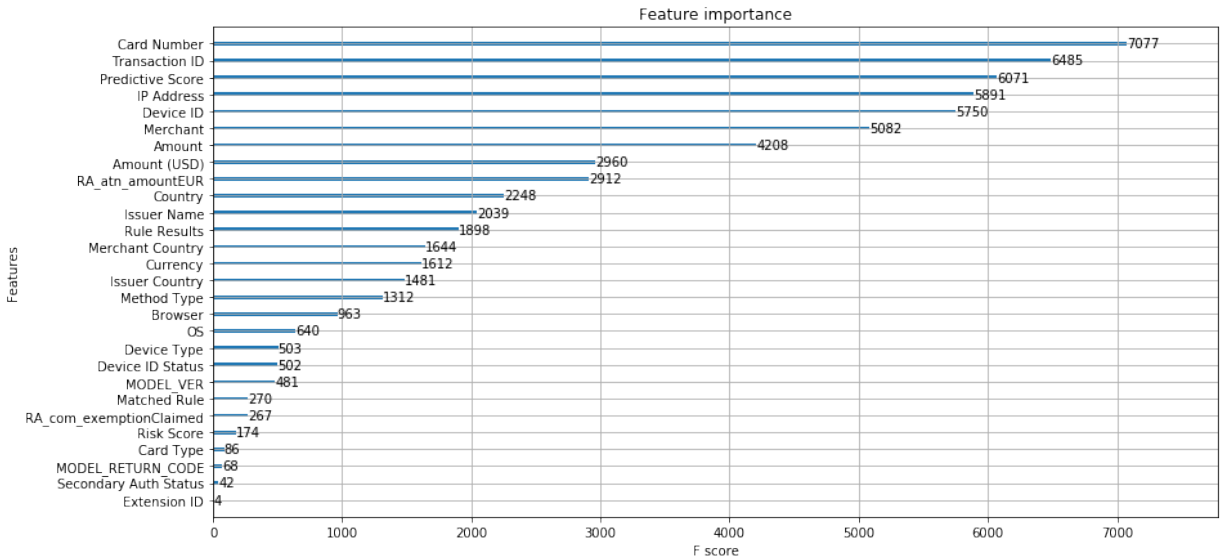


**Fig. 2** XGBoost Feature Importance

attributes in order for the other attributes to contribute meaningful information to the model. For example, let's consider Table 2. If the user with card number '2468' has two specific devices that he normally uses to make purchases ('M987', 'K214') but then uses a device with a different ID, 'C199', then this transaction can be flagged as potential fraud. This is also true for other attributes such as the ones depicted in the Table 2. In addition, it is important to note that a transaction can be flagged if the "Merchant" is one that is highly related to fraud, that is, regardless of the "Card Number".

Table 3 shows the comparison of the sensitivity and FN:TP ratio of the current system, XGBoost with raw transaction data, XGBoost with both raw and aggregated features which were computed for various time frames, and the benefit-based XGBoost classifier with aggregated features computed using the time frame with the best results. Also, as a reminder, the goal of the models is to

**Table 1** Example of a Repeated Fraudulent Transaction

| Card Number | Transaction ID | Merchant | Amount | Transaction Time |
|---|---|---|---|---|
| 1234 | 5498 | TicketsOnline | 10.00 | 10:20:14 |
| 1234 | 5499 | TicketsOnline | 10.00 | 10:22:05 |
| 1234 | 5501 | TicketsOnline | 10.00 | 10:22:55 |

**Table 2** Example of a Fraudulent Transaction Identified using Historical Patterns

| Card Number | Device ID | IP Address | Merchant | Amount |
|---|---|---|---|---|
| 2468 | M987 | 192.168.1.2 | Amazon | 25.87 |
| 2468 | K214 | 170.55.11.10 | CoolKidz | 20.99 |
| 2468 | K214 | 170.55.11.10 | Amazon | 158.42 |
| 2468 | M987 | 192.168.1.2 | CoolKidz | 99.99 |
| **2468** | **C199** | **187.10.10.12** | **SportsZone** | **4250.99** |

improve the sensitivity of the current system and also to achieve a FN:TP ratio of 5:1 or lower.

From Table 3 we see that all XGBoost models had significant improvement over the current system. The XGBoost model with aggregated features computed using 180 days of history gave the best results compared to all other time frames as well as the XGBoost model using raw results. It improved sensitivity by 17.5% compared to the current system. It also achieved a FN:TP ratio of 2:1 compared to 7:1 from the current system. Hence, this time frame was chosen for experiments using the benefit-based XGBoost classifier. The aim of the benefit-based classifier is to determine if we can achieve improved results when taking benefits and costs into consideration when training the model. Since the dataset is highly skewed and also since it is more beneficial to correctly identify fraudulent cases rather than legitimate cases, we need to ensure that the chosen model is as accurate as possible when handling fraudulent cases. From Table 3, we can see that the benefit-based model not only improved the FN:TP ratio (reduced it to 1:1) but also achieved the best sensitivity value out of all the models, 51.2%. Hence, this model was selected for the offline detection system.

In order to ensure that the benefit-based XGBoost model performs satisfactory regardless of the chosen benefit and cost values, sensitivity analysis was performed using a wide range of benefit and costs values which produced $\eta$ values between 0 and 1 (the upper and lower bound for $\eta$). Figure 3 illustrates the benefit scores, $\mu$, achieved for the various values of $\eta$, both by the benefit-based XGBoost and accuracy-based XGBoost models. We can see that the benefit-based model achieved higher benefits in all cases except when $\eta = 1$. When $\eta = 1$, this represents the same benefits and costs as that of traditional accuracy based XGBoost. Hence the benefit scores for both models were the same for this scenario.

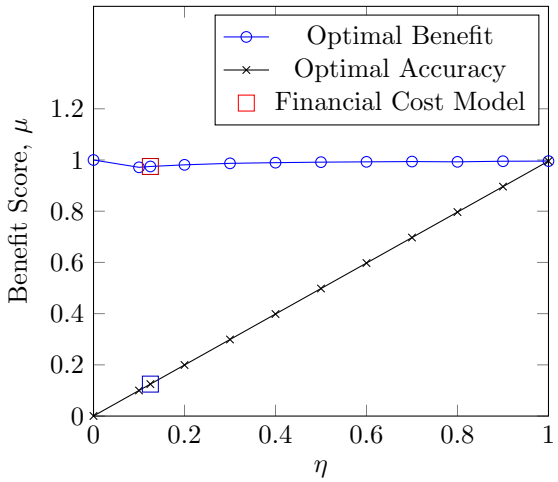


**Fig. 3** Benefit Scores for Benefit-Based XGBoost Classifier

Figure 4 shows the accuracy scores achieved by the benefit-based XGBoost model during the the sensitivity analysis experiments. We can see from the graph that for some scenarios, accuracy was sacrificed slightly in order to achieve improved benefits, that is, better performance with regards to the fraudulent cases.

**Table 3** Sensitivity and False Negative to True Positive Ratios of Current System vs XGBoost Models

| Model | Sensitivity | FN : TP ratio |
|---|---|---|
| Current System | 14.5% | 7:1 |
| XGBoost | 19.5% | 4:1 |
| XGBoost with Aggregated Features: | | |
| - 5 days history | 26.7% | 3:1 |
| - 10 days history | 24.9% | 3:1 |
| - 30 days history | 30.2% | 2:1 |
| - 180 days history | 32.0% | 2:1 |
| Benefit-Based XGBoost | 51.2% | 1:1 |

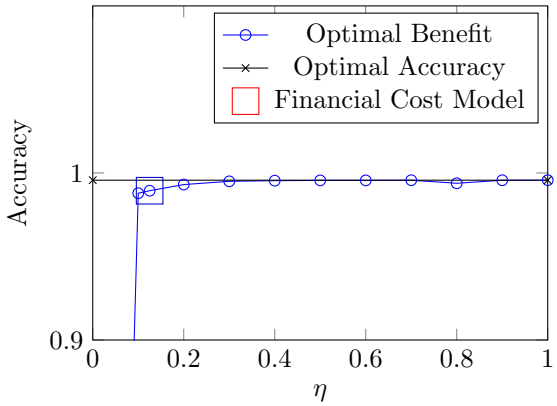


**Fig. 4** Accuracy Scores for Benefit-Based XGBoost Classifier

For the online real-time system, the decision tree from the best iteration of the XGBoost model (using just raw transaction features) was generated. The rules will be extracted and incorporated into the current online system.

## 5 Conclusions

With more businesses moving their business online, credit card fraud is an ongoing problem that needs to be prevented in order to keep customers safe. Fraudsters tend to change up their patterns and customer behavior also changes overtime. Therefore, fraud detection techniques need to be dynamic and efficient, making machine learning algorithms an ideal solution. In addition, credit card fraud datasets are normally skewed towards the positive class (legitimate cases) causing traditional machine learning algorithms to err on the side of the positive class. However, this is not ideal since it is more important to identify fraudulent transactions than legitimate ones. In this paper, we propose a technique for improved credit card fraud detection by first performing data aggregation on customer transactions in order to creative new features based on their periodic data and then by applying these along with the original raw features to a benefit-based XGBoost classifier. The first step handles learning customer spending behaviors and the benefit-based XGBoost classifier accounts for costs and benefits when training so that the model is not skewed towards the majority class but instead focuses on achieving maximum benefits. This research was performed in collaboration a bank whose objective is to achieve a false negative to true positive ratio of 5:1 or lower. The proposed model achieved a ratio of 1:1 thus not only satisfying the requirements of the bank but also achieving excellent performance with the fraudulent cases depicted by the increase in benefits compared to the traditional accuracy approach. For future work, we plan to apply benefit-based learning to other classifiers.

## Declarations

**Funding** No funding was received to assist with the preparation of this manuscript.

**Conflict of Interest** The authors have no relevant financial or non-financial interests to disclose.

**Authors' Contributions** Shellyann Sooklal and Patrick Hosein wrote and reviewed the main manuscript text. Shellyann Sooklal implemented and tested the methods presented in the manuscript and hence prepared Figures 1-4 as well as Tables 1-3.

**Data Availability** The datasets generated during and/or analysed during the current study are not publicly available since these datasets were provided by a private bank and the bank would like to keep any information on themselves and their customers confidential.

## References

[1] Alkhatib KI, Al-Aiad AI, Almahmoud MH, Elayan ON (2021) Credit card fraud detection based on deep neural network approach. In: 2021 12th International Conference on Information and Communication Systems (ICICS), pp 153–156, DOI 10.1109/ICICS52457.2021.9464555

[2] Almhaithawi D, Jafar A, Aljnidi M (2020) Correction to: Example-dependent cost-sensitive credit cards fraud detection using smote and bayes minimum risk. SN Applied Sciences 2:1995, DOI 10.1007/s42452-020-03810-y

[3] Babu AM, Pratap A (2020) Credit card fraud detection using deep learning. In: 2020 IEEE Recent Advances in Intelligent Computational Systems (RAICS), pp 32–36, DOI 10.1109/RAICS51191.2020.9332497

[4] Bahnsen AC, Aouada D, Stojanovic A, Ottersten B (2015) Detecting credit card fraud using periodic features. In: 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), pp 208–213, DOI 10.1109/ICMLA.2015.28

[5] Dash R, Rautray R, Dash R (2021) A legendre neural network for credit card fraud detection. In: Mishra D, Buyya R, Mohapatra P, Patnaik S (eds) Intelligent and Cloud Computing, Springer Singapore, Singapore, pp 411–418

[6] Dileep MR, Navaneeth AV, Abhishek M (2021) A novel approach for credit card fraud detection using decision tree and random forest algorithms. In: 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), pp 1025–1028, DOI 10.1109/ICICV50876.2021.9388431

[7] Jain V, Agrawal M, Kumar A (2020) Performance analysis of machine learning algorithms in credit cards fraud detection. In: 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), pp 86–88, DOI 10.1109/ICRITO48877.2020.9197762

[8] Jain V, Kavitha H, Mohana Kumar S (2022) Credit card fraud detection web application using streamlit and machine learning. In: 2022 IEEE International Conference on Data Science and Information System (ICDSIS), pp 1–5, DOI 10.1109/ICDSIS55133.2022.9915901

[9] Jog A, Chandavale AA (2018) Implementation of credit card fraud detection system with concept drifts adaptation. In: Bhalla S, Bhateja V, Chandavale AA, Hiwale AS, Satapathy SC (eds) Intelligent Computing and Information and Communication, Springer Singapore, Singapore, pp 467–477

[10] K K, B M, K S, D JP, Sree Lakshmi D (2022) Credit card fraud identification using logistic regression and local outlier factor. In: 2022 Second International Conference on Interdisciplinary Cyber Physical Systems (ICPS), pp 99–103, DOI 10.1109/ICPS55917.2022.00026

[11] Khatri S, Arora A, Agrawal AP (2020) Supervised machine learning algorithms for credit card fraud detection: A comparison. In: 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), pp 680–683, DOI 10.1109/Confluence47617.2020.9057851

[12] Krishna M, Praveenchandar J (2022) Comparative analysis of credit card fraud detection using logistic regression with random forest towards an increase in accuracy of prediction. In: 2022 International Conference on Edge Computing and Applications (ICECAA), pp 1097–1101, DOI 10.1109/ICECAA55415.2022.9936488

[13] Lopes AP, Parshionikar S, Kale A, Sharma N, Varghese AA (2021) Comparative analysis of deep learning techniques for credit card fraud

detection. In: 2021 International Conference on Advances in Computing, Communication, and Control (ICAC3), pp 1–5, DOI 10.1109/ICAC353642.2021.9697205

[14] Lucas Y, Portier PE, Laporte L, Calabretto S, He-Guelton L, Oblé F, Granitzer M (2019) Dataset shift quantification for credit card fraud detection. In: 2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE), pp 97–100, DOI 10.1109/AIKE.2019.00024

[15] Madhavi A, Sivaramireddy T (2021) Real-time credit card fraud detection using spark framework. In: Mai CK, Reddy AB, Raju KS (eds) Machine Learning Technologies and Applications, Springer Singapore, Singapore, pp 287–298

[16] Marchal S, Szyller S (2019) Detecting organized ecommerce fraud using scalable categorical clustering. In: Proceedings of the 35th Annual Computer Security Applications Conference, Association for Computing Machinery, New York, NY, USA, ACSAC '19, p 215–228, DOI 10.1145/3359789.3359810, URL https://doi.org/10.1145/3359789.3359810

[17] Minastireanu E, Mesnita G (2019) An analysis of the most used machine learning algorithms for online fraud detection. Informatica Economica 23:5–16, DOI 10.12948/issn14531305/23.1.2019.01

[18] Modi K, Dayma R (2017) Review on fraud detection methods in credit card transactions. In: 2017 International Conference on Intelligent Computing and Control (I2C2), pp 1–5, DOI 10.1109/I2C2.2017.8321781

[19] Nanduri J, Liu YW, Yang K, Jia Y (2020) Ecommerce fraud detection through fraud islands and multi-layer machine learning model. In: Arai K, Kapoor S, Bhatia R (eds) Advances in Information and Communication, Springer International Publishing, Cham, pp 556–570

[20] Robertson (2021) Credit card fraud nilson report. Nilson Report 18

[21] Saddam Hussain SK, Sai Charan Reddy E, Akshay KG, Akanksha T (2021) Fraud detection in credit card transactions using svm and random forest algorithms. In: 2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), pp 1013–1017, DOI 10.1109/I-SMAC52330.2021.9640631

[22] Shah A, Mehta A (2021) Comparative study of machine learning based classification techniques for credit card fraud detection. In: 2021 International Conference on Data Analytics for Business and Industry (ICDABI), pp 53–59, DOI 10.1109/ICDABI53623.2021.9655848

[23] Singh Y, Singh K, Singh Chauhan V (2022) Fraud detection techniques for credit card transactions. In: 2022 3rd International Conference on Intelligent Engineering and Management (ICIEM), pp 821–824, DOI 10.1109/ICIEM54221.2022.9853183

[24] Somasundaram A, Reddy S (2019) Parallel and incremental credit card fraud detection model to handle concept drift and data imbalance. Neural Computing and Applications 31:3–14, DOI 10.1007/s00521-018-3633-8

[25] Sooklal S, Hosein P (2020) A benefit optimization approach to the evaluation of classification algorithms. In: Hemanth DJ, Kose U (eds) Artificial Intelligence and Applied Mathematics in Engineering Problems, Springer International Publishing, Cham, pp 35–46

[26] Srivastava A, Yadav M, Basu S, Salunkhe S, Shabad M (2016) Credit card fraud detection at merchant side using neural networks. In: 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), pp 667–670

[27] Sumanth C, Kalyan PP, Ravi B, Balasubramani S (2022) Analysis of credit card fraud detection using machine learning techniques. In: 2022 7th International Conference on Communication and Electronics Systems (ICCES), pp 1140–1144, DOI 10.1109/ICCES54183.2022.9835751

[28] Tanouz D, Subramanian RR, Eswar D, Reddy GVP, Kumar AR, Praneeth CVNM (2021) Credit card fraud detection using machine learning. In: 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), pp 967–972, DOI 10.1109/ICICCS51141.2021.9432308

[29] Thennakoon A, Bhagyani C, Premadasa S, Mihiranga S, Kuruwitaarachchi N (2019) Real-time credit card fraud detection using machine learning. In: 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), pp 488–493, DOI 10.1109/CONFLUENCE.2019.8776942

[30] Wang C, Han D (2019) Credit card fraud forecasting model based on clustering analysis and integrated support vector machine. Cluster Computing 22:13,861–13,866, DOI 10.1007/s10586-018-2118-y

[31] Wang C, Wang Y, Ye Z, Yan L, Cai W, Pan S (2018) Credit card fraud detection based on whale algorithm optimized bp neural network. In: 2018 13th International Conference on Computer Science & Education (ICCSE), pp 1–4, DOI 10.1109/ICCSE.2018.8468855

[32] Whitrow C, Hand D, Juszczak P, Weston D, Adams N (2009) Transaction aggregation as a strategy for credit card fraud detection. Data Mining and Knowledge Discovery 18:30–55, DOI 10.1007/s10618-008-0116-z

[33] Yeşilkanat A, Bayram B, Köroğlu B, Arslan S (2020) An adaptive approach on credit card fraud detection using transaction aggregation and word embeddings. In: Maglogiannis I, Iliadis L, Pimenidis E (eds) Artificial Intelligence Applications and Innovations, Springer International Publishing, Cham, pp 3–14