Building Efficient Classifiers for Detecting Fraudulent Transactions

Sydney Hu, Caleb Reimer, Allison Martin, Gerard LeBlanc, and Jackson Cone

Topic, Business Justification, and Problem Statement

Background Information:

In 2021 alone, there were over 1.6 million reports of identity theft, including almost 400,000 cases of credit card fraud. In the US, credit card fraud led to over $10 billion in losses in 2020 and this figure is expected to grow to $17 billion by 2030. Moreover, data breaches were up almost 70% in 2021 compared to 2020, which set a new all-time record. With the ever increasing threats to cyber security systems, it is imperative to develop additional safe-guards to protect both consumers as well as credit-card companies from fraud. In contrast to cyber security, which is constantly evolving, credit card and e-commerce transaction data is relatively consistent over time. Many transactions share sets of common features (e.g. purchaser, seller, location, price, time of day). Consequently, developing and refining novel algorithms to detect and reject fraudulent credit card transactions can be a cost-effective tool for reducing losses to consumers and businesses. Here, our goal was to test various methods for detecting fraudulent transactions (credit card and e-commerce). To do so, we used publicly available labeled transaction data that included both valid and fraudulent transactions.

For Americans in particular online fraud has become increasingly prevalent. In 2021 alone roughly 2.8 million people filed a fraud report with an increase of more than 70% from the previous year. Younger Americans were more often targeted for fraud, however individuals over the age of 70 were reported to have lost more money. The average person over age 80 lost $1,500, which is three times for the typical victim in their 20s. This led us to want to focus on online specific fraud cases, and within that discover which groups of individuals are most at risk by looking at different demographic features about the individual and the case. However, due to the fact that the majority of transactions are non-fraudulent, we need models to efficiently identify cases of fraud. Therefore, *<u>we focused on modeling methods that are capable of isolating anomalous data points from the entirety of observations</u>*. After conducting literary reviews, we found that clustering methods would reveal fraud within the banking sector, decision trees could be used for an analytic approach to fraud detection, and isolation forests would best be used for detection of rare events; thus we decided on these methods which would best capitalize on atypical data.

Data

Source: https://www.kaggle.com/datasets/vbinh002/fraud-ecommerce
2 datasets: **Fraud_Data.csv** and **IpAddress_To_Country.csv**
Link: https://www.dropbox.com/sh/atpt6d6hhto13jt/AACMoSFrWqtGmFmQbNyiV1rEa?dl=0

Data Description: The original dataset (Fraud_Data.csv) consisted of over 151,112 e-commerce transactions with 11 variables: A secondary data file (IpAddress_To_Country.csv) was also included with the main data that matched lower and upper bounds of IP address numbers to countries of origin. Interestingly, we found that fraudulent transactions in this dataset account for $523,488 in total purchase value.

- user_id: a unique ID for each purchaser
- signup_time: time user logged into website
- purchase_time: the time the transaction was executed
- purchase_value: the cost of the item
- device_id: an ID corresponding to the computer that the transaction originated from
- source: a categorical variable corresponding to how the purchaser ended up on the host site [SEO (search engine), Ads, Direct]
- browser: a categorical variable corresponding to the browser used for the transaction [Chrome, Opera, Firefox, Safari, etc.]
- sex: a binary variable corresponding to the sex of the purchaser [M (male), F (female)]
- age: an integer corresponding to the age of the purchaser
- ip_address: a floating point value corresponding to the ip address of the purchaser computer.
- class: a binary variable [0 (not fraud), 1 (fraud)]

Data Cleaning and Feature Engineering

Location: Using a second dataset that links ranges of ip addresses with their countries of origin, we wrote a simple search algorithm to link the ip address associated with each transaction with a country of origin and append this information to the main dataframe. Importantly, not all transactions had ip addresses that could be matched to a location. Of the 151,112 transactions, 79,000 transactions were matched to locations. While location information might be useful for identifying fraudulent transactions, it is also possible that transactions that cannot be matched are more likely to be fraudulent, perhaps due to VPNs or other tools for obscuring information exchanged between computer and server. Thus, a location of “None” was entered for transactions that could not be matched.

User_id: Exploratory analyses revealed that the variable “user_id” was unlikely to be informative as there were as many unique user ids as there were observations in the dataset, meaning that no single user id was associated with more than one transaction. Thus, this variable was omitted from consideration.

Device_id: There were 137,956 unique device ids in the dataset. No single id engaged in more than 20 transactions. Unsurprisingly, this variable did not carry much predictive power as was omitted from further analyses.

Purchase_time: Purchase times were stored as character arrays and included the year, month, day, and time of the transaction. All transactions took place in 2015 so the year was irrelevant to analyses. However, transactions occurred every month, day, and hour. We created new variables corresponding to the month, day, and hour of each transaction. We also created categorical variables to lump transactions broadly into segments of the day [morning (6:00 - 12:00); afternoon (12:00 - 18:00); evening (18:00-24:00); night (00:00 - 6:00)].

Purchase_value: The purchase values in the dataset range from $9 to $154. The mean purchase value was $36.94 ± $18.32 SD. We also created new variables to treat price as a categorical variable [$20 or less, $20-50, $50-100, greater than $100].

Age: The ages in our dataset ranged from 18 to 76 with a mean age of 33.14 ± 8.62 years (SD). In addition to treating age as a continuous variable, we also lumped age ranges together according to commonly used age groups. The age ranges are as follows: [18-24; 25-34; 35-44; 45-54; 55-64; over 65].

Sex: Of all transactions that were completed by males, 9.5% were fraudulent. Similarly, 9.1% of all transactions that were completed by females were fraudulent. These proportions are very similar to the overall level of fraud across the dataset (~9.3%) suggesting that the Sex of the purchaser will not provide much predictive value in our models.

Signup_purch_delta: The dataset contained information about the time of purchase as well as the time that the purchaser signed up for a website. We computed a new variable related to the time between signup time and purchase by converting the date strings to a datetime and then subtracting the two and dividing by 60 to get the total delta in minutes.

Exploratory Data Analysis

We first sought to determine whether any individual variables in our dataset were associated with fraud because this could help us decide which variables could be most useful to include in our models. Several variables didn't appear to have any links to fraudulent transactions. The distributions of purchaser age and item value are highly overlapping (Figure 1), indicating that specific ages or item prices do not appear to carry information for predicting the probability of a fraudulent transaction.

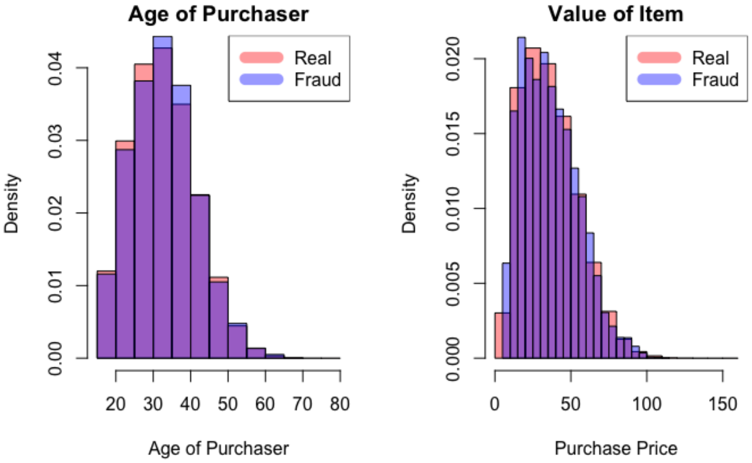


Figure 1. The histograms show the relative proportions of real (red) versus fraudulent transactions (blue) associated with different purchaser ages (left) or item values (right).

We found similar results when we examined information about the source of the transactions (e.g., Geographic Location (determined by matching ip addresses), how the user ended up on the website (SEO, Ad, Direct) or the Browser the purchaser was using when the purchase was made. The counts of transactions made on different browsers for real transactions versus fraudulent transactions revealed that the relative proportions are similar for the two transaction types (data not shown). Moreover, a similar pattern is observed when considering how the purchaser came to arrive on the sale website. Furthermore, we also examined whether

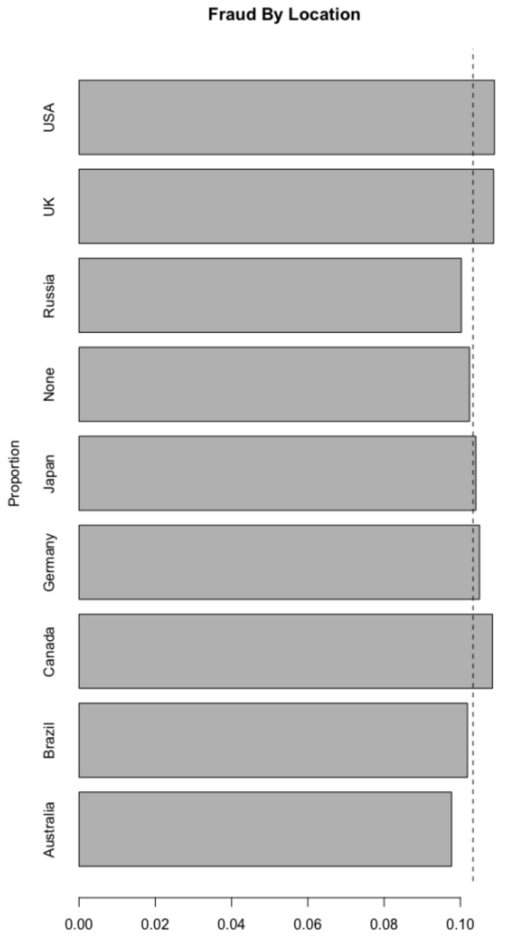


Figure 2. Bar plot depicts the proportion of frauds in the top 9 most common locations. Dashed line depicts the overall proportion of fraud across the entire e-commerce dataset.

particular locations were more likely to be associated with fraud. We found that 9 of the 10 most common locations for real transactions matched the top 10 locations for fraud. In addition, the overall proportion of fraudulent transactions across these locations were closely aligned with the overall proportion of fraud across the entire dataset (Figure 2, dashed line), suggesting that frauds were unlikely to be clustered in particular countries of origin. Taken together, these data indicate that the variables related to location, source, or browser are unlikely to carry much predictive power on their own. However, it is possible that combining information contained in these variables could be useful so we still considered them for use in our initial multivariate models.

Lastly, there were other variables that did appear to be associated with fraudulent transactions. We found that fraudulent transactions were overrepresented in the first month of the year compared to real transactions. Days in the early part of each month also tended to be more likely to be fraudulent. Perhaps the most striking result of our exploratory analysis resulted from examining the variable we created that tracked the time delay between when a user signed up for a website and when the transaction took place. Comparing the distribution of these values for real and fraudulent charges, showed that a disproportionate number of fraudulent charges occurred almost immediately after a user signed up, often just seconds or minutes after (Figure 3). This may be indicative of bots creating accounts and then immediately attempting to make purchases. This variable will likely be useful for identifying fraudulent transactions.

Overview of Modeling

Models Tested:

<u>Isolation Forests</u> can be used to identify data points that are somehow distinct from the bulk of the other data points (anomaly detection). It works by randomly splitting the data across different trees and then asking how many splits are required on average to isolate particular data points. The primary output of the isolation forest is the average depth of each point (the average number of splits required for isolation). Related to this is the anomaly score, where the higher the anomaly score, the fewer splits were required to isolate. The average depth or anomaly score can be used to predict the class of a data point by assigning a threshold value of depth or score to assign a point to the anomalous class or not.

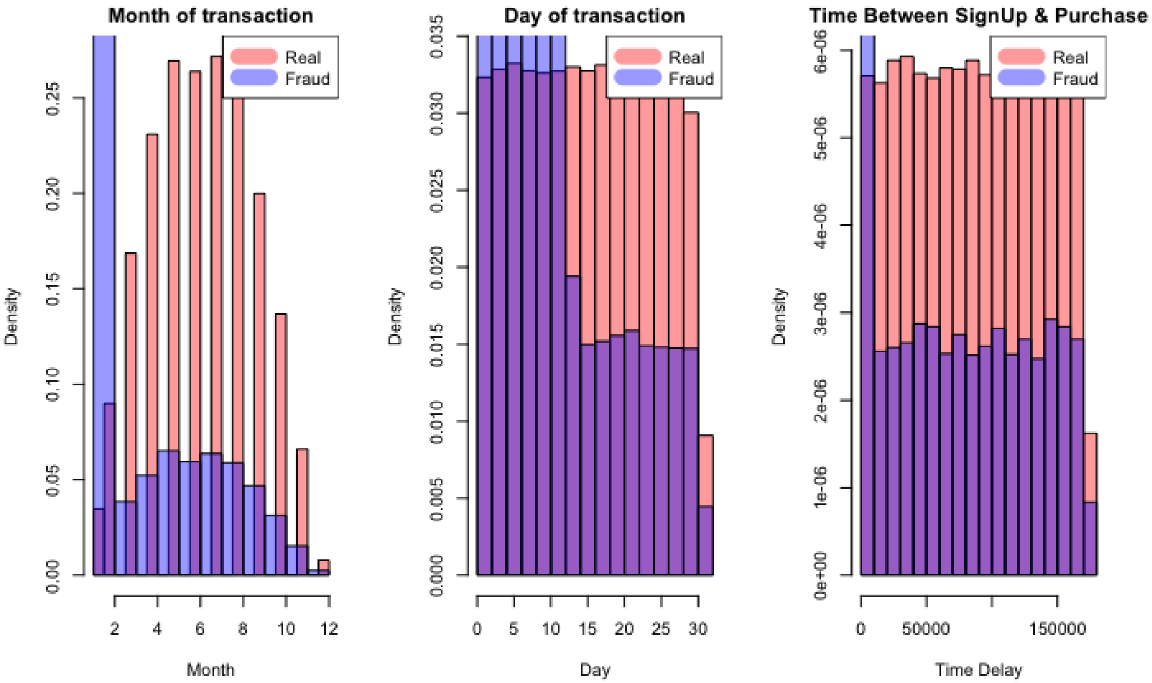


Figure 3. Histograms depict the proportion of fraudulent (blue) transactions versus real (red) transactions by month of the year (left) or day of the month (middle). Right histogram depicts the time delay between when a user signed up for a website and when the transaction took place.

<u>Gradient Boosting machine (GBM)</u> is a tree based model. It works by selecting a set of features for each tree to find the best splits. At every iteration of the tree building process, the model uses information where prior trees performed poorly to improve the performance of the next model. The notion is that the model learns where it performs poorly and uses those insights to ultimately remove as much error as possible.

<u>One-Class Support Vector Machine (OC-SVM)</u> is an unsupervised learning algorithm which is used in order to create distinct samples with the use of a response variable by learning by training from single class samples. This technique is widely used as a solution at anomaly detection to identify outliers in the data.

Model Selection: Prior to fitting, we split the dataset into training (60%), validation (20%), and test (20%) sets. All models were trained on the training data and their performance was compared by testing their prediction accuracy on the validation data. The highest performing model was selected based on performance on the validation. The trained final model of each model type (GBM, isolation forest, SVM) were then compared by testing their prediction accuracy on the test data.

Hyperparameters: Hyperparameter tuning was performed by fitting different models with various combinations of hyperparameters and then testing the prediction performance of each variant on the validation data. The hyperparameters that yielded the best prediction performance were then used for performance comparisons with the other model types.

Isolation Forest Results

We first attempted to fit isolation forest models on subgroupings of different variable types to determine whether any of these variable combinations could be useful for detecting fraud. Our initial groupings focused on three areas:

- Source of Transaction {Location, Browser, Source [SEO, Ad, Direct]}
- Transaction Information {Purchase Value, Age, Sex}
- Time of Transaction {Month, Day, signup_purch_delta}

As suggested by our initial exploratory analysis, isolation forest models fit to Source variables alone (*Location, Browser, Source*) were not very useful for predicting fraud (Figure 4, top). There were no evident differences in the average depth needed to isolate real versus fraudulent data points when only these variables were considered, suggesting Source variables do not contain much predictive power. We found similar results when fitting isolation forest models exclusively using variables associated with information about the transaction (*Purchase Value, Age, Sex,* Figure 4, middle). These modeling results were consistent with our exploratory analysis and indicate that combining information across these classes of predictors does not add additional predictive power.

We then fit an isolation forest model using information about the time of transaction (*Month, Day, signup_purch_delta*). Unlike our prior models, there appeared to be a difference in the depth of data points associated with real versus fraudulent transactions (Figure 4, bottom). This suggests that time variables may be useful for

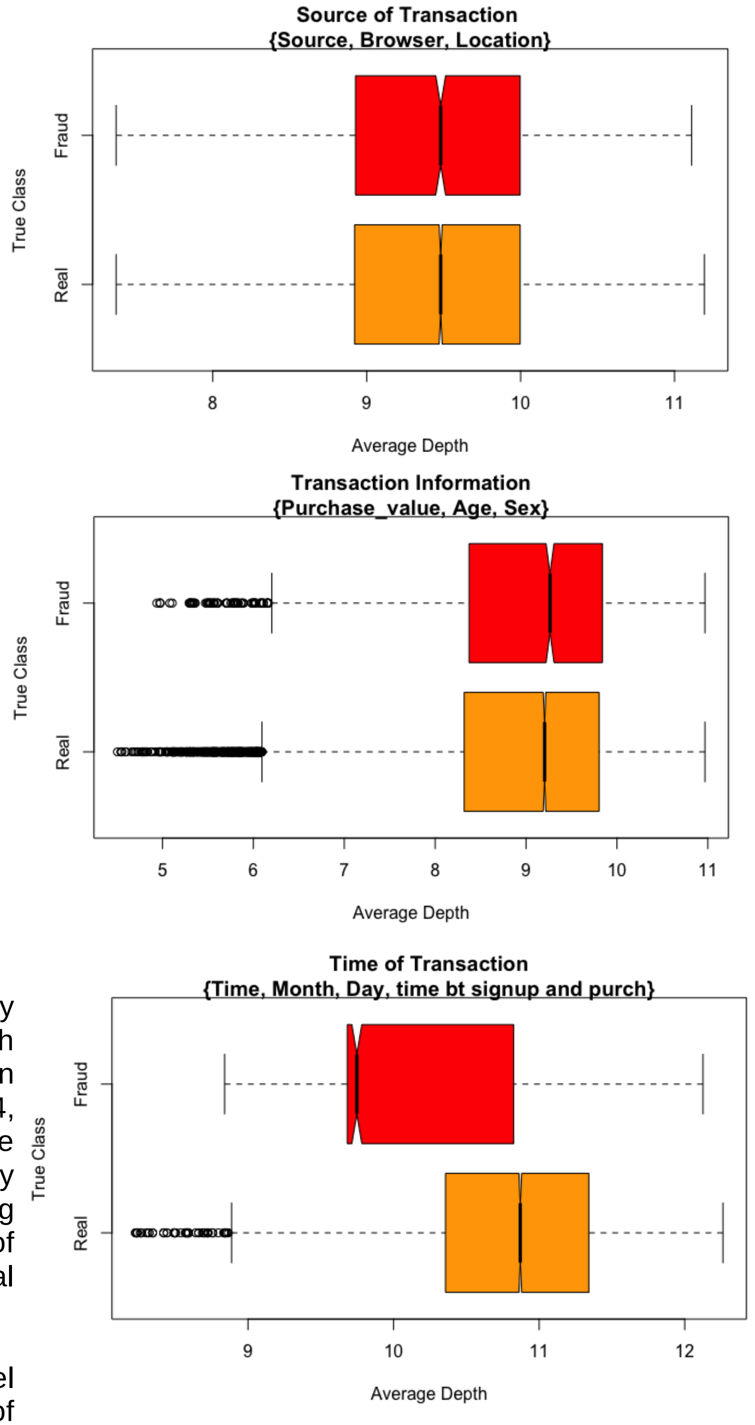


Figure 4. Box plots depict the average depth for the training data for real (orange) versus fraudulent (red) transactions when Source variables are used to fit the model (top), transaction variables (middle), or time variables (bottom).

prediction. We also attempted converting information about time of day, age, and purchase value into categorical variables but the primary results were unchanged.

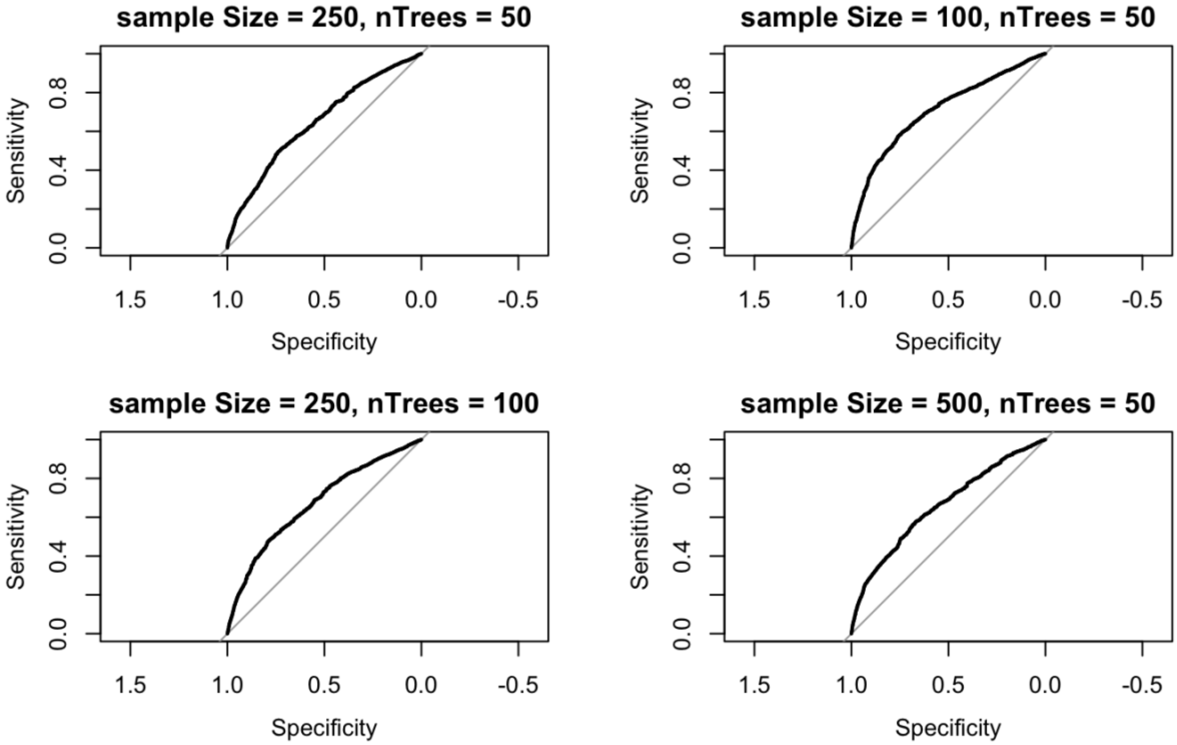


Figure 5. Image shows the ROC curves for four models with different input parameters (listed at the top).

Given the results of our preliminary models, we then fit a new isolation forest model using *month, day, signup_purch_delta, location, source, and browser* as predictors. We experimented with different hyperparameters such as the number of trees in the forest (*nTrees*) and the number of data samples that are used to build each tree (*sample size*). The models were fit to the training data and the Area Under the Receiver Operating Characteristic (AUROC) was computed after predicting whether the validation data points were fraudulent or not (Figure 5). The model on the top right (sample size = 100; nTrees = 50) produced the largest AUROC value (0.71) when predicting the class of the validation data of any of the models.

The best model showed a strong relationship between the predicted average depth of the validation data points and whether or not those points were fraudulent transactions (red) or legitimate (orange). Note, that while Source variables (*Location, Browser, Source)* were not strongly associated with average depth in our preliminary models, including them together with Time variables reduced the variance in the average depth of predictions. For example, omitting the Location variable from the best model reduced the AUROC from 0.7 to 0.65.

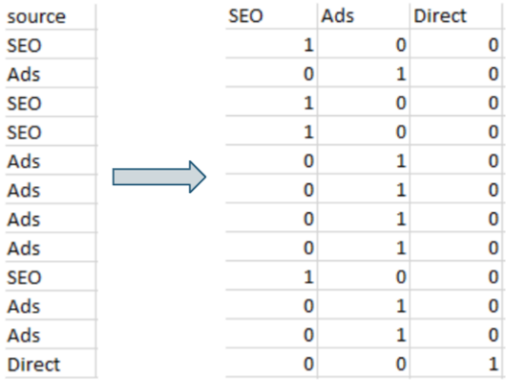
To compare the prediction accuracy of the best isolation forest model, we then used it to predict the class (fraud vs. real) of the held out test data (Figure 6). Interestingly, we found that the best model had both appreciably high false positive and false negative rates (Sensitivity = 0.67, Specificity = 0.65). Thus while the model performed well (AUROC ~0.7), the model failed to correctly classify many fraudulent transactions and misclassified many legitimate transactions as fraudulent. This is problematic for two reasons:

1. Declining legitimate sales hurts revenue and could discourage customers from purchasing in the future.
2. Many fraudulent transactions would be allowed to occur, which could impact business not only due to product and person-hour losses but also create issues with companies that insure such losses.

To estimate the losses associated with this model, we computed the average cost of legitimate and fraudulent sales in the dataset (both ~$30). We then multiplied the costs by the number of false negatives (969, the number of fraudulent transactions that would be allowed to go through) or the number of false positives (1848, the number of legitimate sales that would be declined as fraud) and added them together. This revealed that the final isolation forest model would yield approximately $366,179 in losses on the test data alone. One positive aspect of this model is that it took less than 1 second to train and test it on held out data, meaning it could be deployed in real-time during online checkouts.

Isolation Forest: Factorized Approach

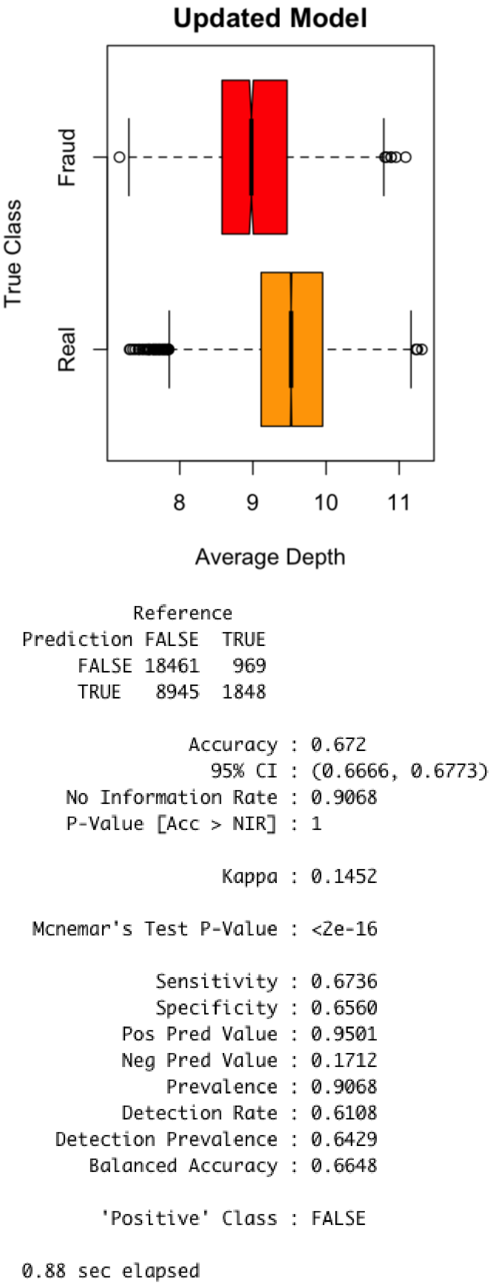
Because our initial attempts of creating an isolation forest model delivered sub-ideal sensitivity and specificity, we decided to perform an input transformation through boolean factorization as shown in figure 7.



source
SEO
Ads
SEO
SEO
Ads
Ads
Ads
Ads
SEO
Ads
Ads
Direct

SEO	Ads	Direct
1	0	0
0	1	0
1	0	0
1	0	0
0	1	0
0	1	0
0	1	0
0	1	0
1	0	0
0	1	0
0	1	0
0	0	1

Figure 7. Augmenting categorical data within a variable into multiple individual variables weighted by boolean value.



```
          Reference
Prediction FALSE  TRUE
     FALSE 18461   969
     TRUE   8945  1848

               Accuracy : 0.672
                 95% CI : (0.6666, 0.6773)
    No Information Rate : 0.9068
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1452

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.6736
            Specificity : 0.6560
         Pos Pred Value : 0.9501
         Neg Pred Value : 0.1712
             Prevalence : 0.9068
         Detection Rate : 0.6108
   Detection Prevalence : 0.6429
      Balanced Accuracy : 0.6648

       'Positive' Class : FALSE

0.88 sec elapsed
```

Figure 6. Summary of final isolation forest model on test data predictions. Boxplot (top) shows the predicted average depth for fraudulent transactions (red) versus real (orange). Confusion Matrix for the final model (bottom).

Data which could be cleanly segmented, such as day of the month and month of the year, were factored into quartiles. Imputed location data, which we believed to be insignificant, could not be easily segmented into less than 7 tranches and was therefore excluded from the final data set to minimize the total number of factors available.

The solitude and ranger packages in r were systematically used to build an isolation forest consisting of 1000 individual trees. This analysis also provided our team with a predicted anomaly score for each transaction. An anomaly score is a measure of the difference in tree depth for an individual tree compared to a random subset of the forest. Because fraudulent transactions are the anomalies in our data set, the number of data splits (tree depth) to classify fraudulent transactions should stick out from the average of the random selection. An anomaly score closer to 1 is more likely to be an anomalous point, and an anomaly score closer to 0 is more likely to be a regular point. Therefore, anomaly scores are inversely related to tree depth.

Figure 8 shows a mapping of the 36 relational dimensions into a 2 dimension scatter plot with each point weighted by anomaly score. Unfortunately, no clear visual relationships were established between these two dimensions which ended this analysis segment.

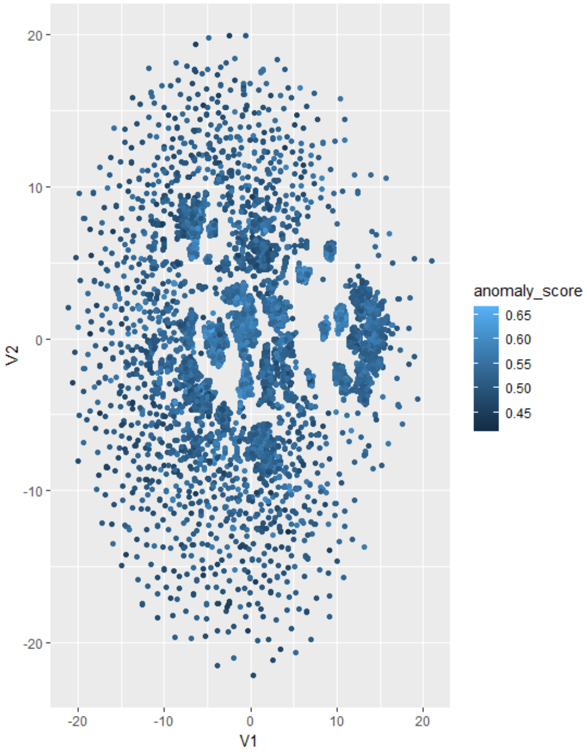


Figure 8. Uniform Manifold Approximation and Projection of the factorized isolation forest model.

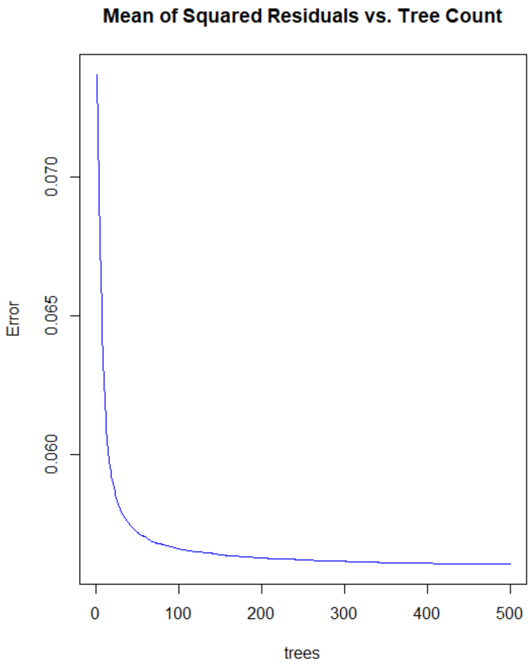


Figure 9. Mean of squared residuals plateaus after approximately 100 trees.

To complete the isolation forest analysis, a new model was built using the randomForest package in r utilizing the factorized dataset with 100 trees. 100 trees were selected for the forest size as this minimized model training run time while remaining near the minimum of the mean of squared residuals compared to a 500 tree forest as shown in figure 9.

Node purity is a measure of increase in mean squared residual, or error, in the forecasting model when a select variable is removed from or permuted within the data set. A higher measure of node purity for a variable therefore indicates a greater level of importance for correct point classification. Figure 10 ranks each variable in order of node purity.

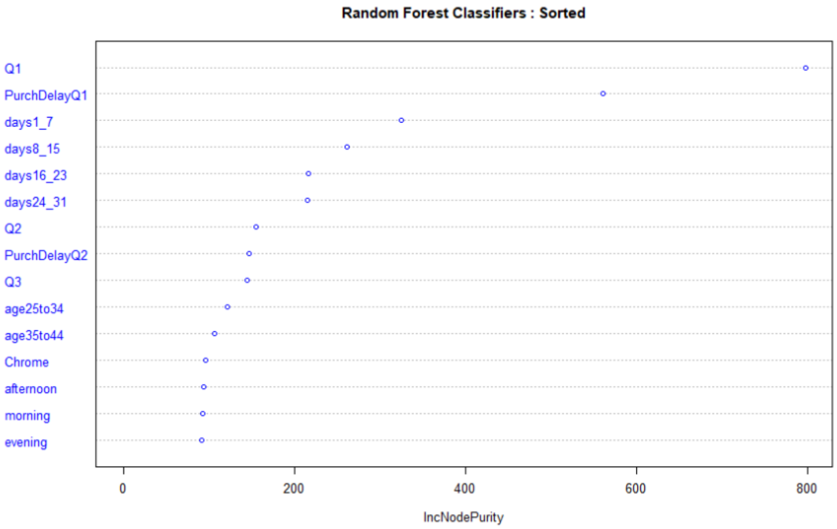


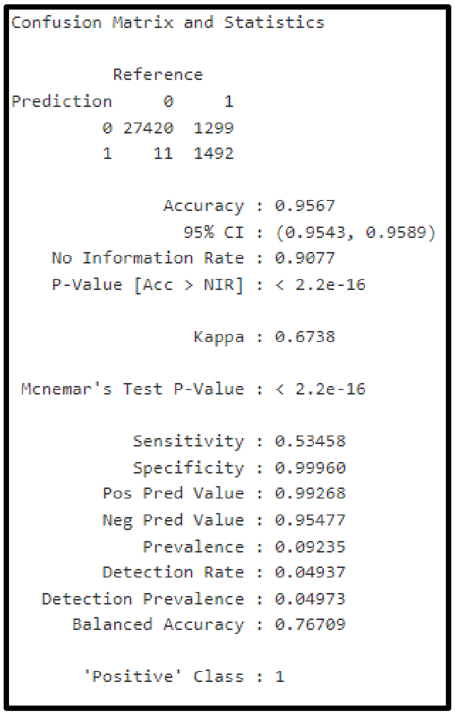
Figure 10. Transactions in the first quarter of the year and transactions made within the first quartile of purchase time delay have the highest node purity and importance for classification.

Gradient Boosting

The first step in building the GBM model was to select features from our data. Many models were created with differing features, and due to the smaller number of attributes, a model including all available features was also tested. Some models focused more heavily on categorical features whereas others focused on raw input variables. The final model had a combination of the two. The final parameters were the signup_purch_delta, source, browser, sex, purchase_value, age, month, day, and categorical variables for the time of day. The categorical features for the age and purchase value were dropped in favor of the raw values to produce a simpler model with comparable results. Almost all variables were included in the final model except for the device id, user id, and raw time values. The location was also not used; although it seemed initially like it may be helpful, the purchase location was not able to be utilized. Due to the large number of unique location values, not all locations were included in the training of the model; therefore, when new locations appeared in validation data sets, an error would occur. This is an exception that could be handled if proved worthwhile but given time considerations the team elected to remove the variable from the model. Although other parameters were included, the signup_purch_delta still dominated the variable importance.

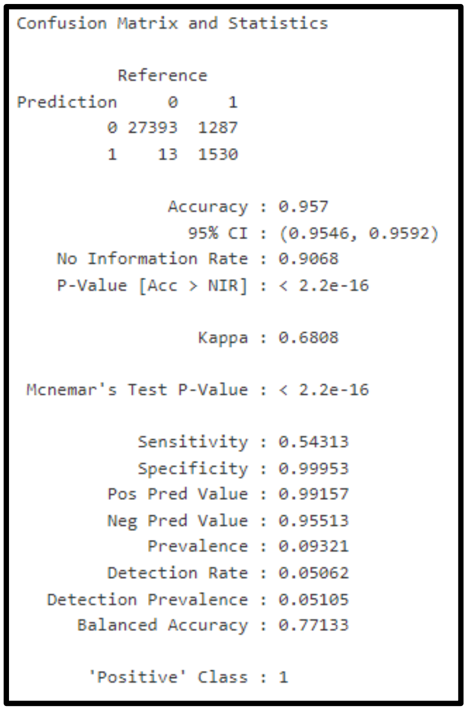
After selecting the optimal features, the next step was to train the final model and then tune hyperparameters. GBM models are generally sufficient at handling imbalanced data and one of the reasons this model was explored to begin with. However, when training the models on the full data set, the results were not encouraging. The model was inclined to predict all observations as non-fraudulent in the validation data set. To counteract this, the data was downsampled to have equal portions of fraudulent and non-fraudulent observations. Surprisingly, this did not have a substantial increase in the number of false positives but significantly increased the number of detected fraudulent charges. Downsampling the training set had the biggest impact on performance. The distribution chosen for the model was Bernoulli due to the classification nature of the problem. The other hyperparameters to investigate were the number of trees, shrinkage, interaction depth, and minimum observations in a node. A range of suitable numbers for all were examined; however, the only hyperparameter that improved performance from the default values was the number of trees, reducing from the default of 500 to 100. The results were the same for the two but with increased training speed for 100 trees. This was expected since GBM hyperparameters require more tuning when dealing with small data sets.

The final model is one that is conservative at predicting fraud but still does not result in excessive false positives. The model has a sensitivity of 0.535 and a specificity of 0.999 when predicting on the validation data set. The full output from the confusion matrix and associated metrics can be seen below. In practical terms, the model correctly predicts over half of the fraudulent charges while falsely identifying a non-fraudulent charge as a fraudulent one only once out of one thousand. One potential drawback for GBM models is that they are prone to overfitting. In our case, the test results were very close to the validation results with a sensitivity of .543 and maintaining a specificity of .999. With similar test results to the validation results, we can expect the model to perform similarly in practice. However, it will be important to continually track performance to ensure this assumption holds and retrain the model if circumstances change. This is especially true since our data set is from 2015, and as we have mentioned, cybersecurity and online commerce is a constantly changing landscape.



```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 27420  1299
         1    11  1492

               Accuracy : 0.9567
                 95% CI : (0.9543, 0.9589)
    No Information Rate : 0.9077
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6738

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.53458
            Specificity : 0.99960
         Pos Pred Value : 0.99268
         Neg Pred Value : 0.95477
             Prevalence : 0.09235
         Detection Rate : 0.04937
   Detection Prevalence : 0.04973
      Balanced Accuracy : 0.76709

       'Positive' Class : 1
```

Validation



```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 27393  1287
         1    13  1530

               Accuracy : 0.957
                 95% CI : (0.9546, 0.9592)
    No Information Rate : 0.9068
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6808

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.54313
            Specificity : 0.99953
         Pos Pred Value : 0.99157
         Neg Pred Value : 0.95513
             Prevalence : 0.09321
         Detection Rate : 0.05062
   Detection Prevalence : 0.05105
      Balanced Accuracy : 0.77133

       'Positive' Class : 1
```

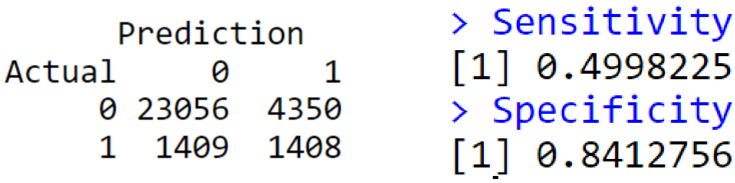
Test

Figure 11. The R outputs showcase the validation and test results for the Gradient Boosting model.

One-Class Support Vector Machines (OC-SVM)

Our team also implemented One-class SVM in an attempt to distinguish the samples by our response variable which is class (fraud/non-fraud). As mentioned in the introduction, since the majority of transactions are normal and few are fraudulent, we chose to test OC-SVM's in order to identify the outliers (fraud cases) presented in the data using anomaly detection. Given our naturally imbalanced dataset, this binary classification algorithm attempts to learn from the training dataset then evaluate the testing set.

On the first attempt at implementation of OC-SVM we found that the training time scaled poorly given the amount of sample data we had; thus the runtime took days to complete. However, this long runtime was solved using data factorization which produced a result in a substantially less amount of time, completing within less than a minute. Nevertheless, the results of running the OC-SVM gave poorer results in terms of sensitivity and specificity than that of GBM, therefore we determined this model would not give us the most optimal results (Figure 12, below).



```
         Prediction
Actual       0     1
     0 23056  4350
     1  1409  1408
```

```
> Sensitivity
[1] 0.4998225
> Specificity
[1] 0.8412756
```

Figure 12. The R outputs showcase the confusion matrix and lower sensitivity and specificity of the One-Class SVM model outputs.

Results and Recommendations

At the advent of our project, our initial goal was to minimize type II errors due to the high cost of letting fraudulent charges continue. The more we performed our analysis, the more evident it became that this is a balancing act. Rejecting legitimate transactions also has a cost associated with it; therefore, we focused our efforts on minimizing losses by taking both into account. The isolation forest model was where we initially focused most of our efforts and spent much time trying to tune. It proved to be an aggressive model and was relatively successful at predicting fraud in a highly imbalanced data set. By our initial hypothesis, this would have been the optimal model to choose. When we consider the large number of rejected transactions, the financial impact becomes larger than with a more conservative model.

The gradient boosting model's results on the test set would have only incurred $48,409 in total losses from both accepted fraudulent transactions and rejected valid transactions. Comparing this with the $366,179 in losses from the isolation forest model, the team recommends implementing the gradient boosting model in the company's transaction operations. The model trains in mere minutes and is able to return predictions immediately making it practical for real-world applications. To improve the user experience even further, an additional step, such as two-factor authentication, could also be utilized to ensure valid charges are not rejected when the model suspects fraud. The team expects this model to be able to significantly reduce the $523,488 in losses observed from the full data set when implemented in future years.

Works Cited

Credit Card Fraud and Detection Techniques: A Review. Delamaire, Abdou, Pointon. Banks and Bank Systems, Volume 4, Issue 2, 2009.

Daly, Lyle, and Jack Caporal. “Identity Theft and Credit Card Fraud Statistics for 2022.” *The Motley Fool*, The Ascent by The Motley Fool, 21 Sept. 2022, https://www.fool.com/the-ascent/research/identity-theft-credit-card-fraud-statistics/.

Detecting Credit Card Fraud by Decision Trees and Support Vector Machines. Sahin and Duman, IMECS 2011.

Duchi, J. (n.d.). Boosting. CS229 Supplemental Lecture notes. Stanford University. http://cs229.stanford.edu/extra-notes/boosting.pdf

Extended Isolation Forests for Fault Detection in Small Hydroelectric Plants, Santis and Costa, MDPI July 2020.

Federal Trade Commission. (2022, February 22). *New data shows FTC received 2.8 million fraud reports from consumers in 2021*. Federal Trade Commission. Retrieved November 17, 2022, from https://www.ftc.gov/news-events/news/press-releases/2022/02/new-data-shows-ftc-received-28-million-fraud-reports-consumers-2021-0

Iacurci, G. (2022, February 22). *Consumers lost $5.8 billion to fraud last year - up 70% over 2020*. CNBC. Retrieved November 17, 2022, from https://www.cnbc.com/2022/02/22/consumers-lost-5point8-billion-to-fraud-last-year-up-70percent-over-2020.html

“Nilson Report.” *Nilson Report | News and Statistics for Card and Mobile Payment Executives*, Dec. 2021, http://nilsonreport.com/.