Comparative Study on Classifying Anomalous Card Transactions with Decision Trees, Random Forest, and SVM using Various Sampling Algorithms

By

Sherranette O. Tinapunan

Professor Nasrin Khansari

Data 698 Research Project

CUNY School of Professional Studies

November 12, 2022

Table of Contents

Abstract

This research utilized anonymized real card transaction dataset from a small business organization to study the performance of Decision Trees, Random Forest, and SVM classification models using various sampling algorithms in classifying anomalous transactions. The sampling algorithms that were considered include undersampling, a combination of undersampling and oversampling, and synthetic oversampling techniques – RWO, SMOTE, MWMOTE, and ADASYN. In addition, the transaction dataset was explored to understand data patterns related to anomalous and normal behaviors. The data was transformed to create metrics that could be utilized as features in the models. These features are then validated by the various models. Finally, a recommendation is made to the small business organization as to which models and sampling techniques should be investigated more for further future work.

Introduction

Card fraud is a worldwide problem that impacts everyone, including consumers, merchants, and card issuers. In 2019, losses due to credit card fraud reached $28.65 billion throughout the world (Lee). More than a third of the total global loss in credit card fraud is in the United States. A Nilson report says that for the next decade, the card industry is projected to incur a global loss of $400 billion in card fraud (Mullen). By 2030, it's projected that the United States is expected to reach a total loss of $17 billion in card fraud with a total card volume of $19 trillion (Mullen). In 2019, the cost for each dollar lost to fraud was $3.13, which include costs associated to expenses such as chargeback fees, restocking merchandise, investigations, legal prosecutions, and security ("Credit Card Transaction Fraud").

There are different ways that card fraud can be perpetrated such as simple theft, counterfeit cards, application fraud, and online fraud. According to Visa, online fraud account for approximately 50% of card fraud in European countries (Sahin and Duman). To help solve this problem, companies are looking into technological solutions. In a statement to CNBC, Mike Lemberger, Visa's senior vice president and regional risk officer for North America, said that through Visa's AI technology, they were able to stop $25 billion in fraud in 2020 (Lee). As you can see, AI solutions in anomaly detection is clearly an important aspect in solving this problem.

This research project was granted permission to utilize anonymized card transaction dataset from a small business organization. For security reasons, the source of the dataset is undisclosed. The objective of this research is twofold. First, to explore the data and gain insights about patterns related to anomalous transactions and to build meaningful metrics that could be utilized as features for the models. Secondly, to investigate the performance of various classification models with different sampling techniques that address the imbalanced nature of the distribution of the anomalous transactions. The insights gained in this research can be used as a starting point for future work recommendations with this organization.

Literature Review

Card fraud is a relevant problem that can be alleviated by machine learning technology. Researchers are benchmarking various algorithms and models and developing techniques to better solve this problem. One such study is published in a paper entitled *Credit Card Fraud Detection – Machine Learning methods* by Varmedja, et al. This study used the Credit Card Fraud Detection dataset published by Kaggle, which contains anonymized credit card transactions made in two days by European cardholders. This dataset contains 284,807 transactions, of which 492 are labeled as fraud (Kaggle). As you can see, the dataset is significantly imbalanced, with only 0.172% fraud cases. This study compared the performance of classical machine learning methods such as Logistic Regression, Random Forest, and Naïve Bayes to deep learning algorithms such as multilayer perceptron and artificial neural network. The study found that classical machine learning methods can perform as well as deep learning algorithms. Classical algorithms such as Random Forest can provide comparable results as a simple neural network (Varmedja, et al.).

In the study, SMOTE (Synthetic Minority Oversampling Technique) was used to address the inherent imbalanced nature of the dataset, and a feature selector tool by Will Koehrsen was used to determine important features. In addition, scaling was done to normalize the magnitude of the features. The metrics recall, precision, and accuracy were used to compare the different models (Varmedja, et al.).

Another study published in a paper entitled *Detecting Credit Card Fraud by Decision Trees and Support Vector Machine* by Sahin and Duman compared the performance between decision trees (DT) and support vector machine (SVM). This study is one of the first studies that compared DT and SVM with a real dataset (Sahin and Duman).

The study was given permission to use a national bank's credit card data warehouse, which contained 978 fraudulent cases and 22 million normal ones. This dataset was highly imbalanced. Stratified sampling was used to undersample the set into meaningful numbers based on the important features determined by their analysis. The types of variables in the data set include transaction statistics, regional statistics, daily amount statistics, and daily number of transaction statistics. For the models to discriminate between fraud and genuine transactions, the stratified samples of genuine records were incrementally combined with fraud transactions in varying ratios. There were three sample sets with 1:1, 1:4, and 1:9 ratio of fraud to legitimate transactions (Sahin and Duman).

The various types of DT methods investigated by the study included CART, C5.0, and CHAID. They also investigated different SVM methods, which included linear, sigmoid, and RBF. The study found that DT models performed better over test data sets but did not do as well on training data sets. This is an indication that SVM overfit the training set; however, as the number of training data increases, the overfitting behavior reduced, and performance became comparable to DT. Accuracy was used to compare the models; however, the study acknowledges that this performance metric is not well matched for this problem domain (Sahin and Duman).

My research project investigated Decision Trees (CART, C5.0, Boosted C5.0), Random Forest, and SVM (linear and RBF). Based on the research by Varmedja, et al., Random Forest show good results with performance comparable to simple neural networks. In addition, the research by Sahin and Duman found that Decision Trees perform better than SVM with smaller training data sets. My research aims to investigate if similar results are observed with the performance of Random Forest compared to the other models and the performance of Decision Tree models compared to SVM.

Research Methodology

The dataset utilized by this research is based on real card transaction data generated by a proprietary application. As mentioned earlier, the source of this dataset is undisclosed for security reasons. The raw dataset contains 304,260 transactions, of which 2,291 are classified as *anomalous*. The more general term, *anomalous transaction*, is the preferred terminology because known *fraudulent* transactions in the data set were augmented by atypical transactions that may exhibit unusual patterns when compared to legitimate transactions. The augmentation includes known test transactions that may exhibit high-volume activity and other transactions related to application failures. The reason for broadening the scope of anomalous transactions is to provide visibility for the other types of transactions that may not necessarily be fraud but are not part of the normal, legitimate customer transactions.

The data preprocessing portion of this project was challenging and is broken down into several steps, with each step resulting in an output file that is fed onto the next layer of data processing. The first step was to go through reports and logs of known fraud cases along with identifying reported transactions associated with known application failures. It is worth noting that the fraud cases experienced were related to fraud card testing activities. In addition, internal, legitimate test transactions were also added to this list. Relevant transaction records were pulled directly from the source database and processed to mark the anomalous transactions. The data was then anonymized. Any internal ID references to contact or transaction records were replaced, encoded, or dropped. Data related to location or application sources were codified.

To reduce the number of missing data that may be significant in the feature selection process, information available in archived file downloads, along with current file downloads from the credit card payment gateway were incorporated into the data set before performing an imputation of the remaining missing data. Metrics were then generated from the raw transaction information, which includes metrics such as the number of distinct card numbers (only the last four digit), the number of distinct IP addresses associated with the customer, daily and yearly transactional amount and volume, time elapsed between transactions, the account age of the customer at the time of purchase, and the daily mean wait time before the next transactions is generated, among other things. Categorical data with high cardinality were meaningfully grouped together to reduce the number of groups. Finally, the data exploratory process reviewed the distribution of the response and predictor variables, near zero variance, correlation, multicollinearity, outliers, and any missing data.

The primary objective of this research is to study the performance of different classifiers and different sampling algorithms to detect anomalous transactions that are part of a private dataset and develop metrics that could be used as features in the models. This research investigated Decision Trees (C5.0, C5.0 boosted tree, CART), Random Forest, and SVM (linear, radial) as classifiers. The sampling algorithms that were considered for each of the classifier were undersampling, combined undersampling and oversampling, RWO (random walk oversampling), SMOTE (synthetic minority oversampling technique), MWMOTE (weighted minority oversampling technique), and ADASYN (adaptive synthetic). The packages *Imbalanced* and *ROSE* were used.

To generate the training-testing set, a stratified sampling of the majority class based on the system source is randomly selected from the entire dataset. The system source identifies the originating application that generated the transactions. This approach to stratify the selection of the majority class by system source was guided by domain knowledge of the general characteristics and patterns of transactions that originate from these sources. The minority class transactions are then combined with the majority class. This data set is then split into 80:20 ratio to use for training and testing.

Data and Methods

The dataset used in this research comprises of 304,260 anonymized credit card transactions, of which 2,291 are classified as anomalous. Figure 1 below shows the daily number of transactions associated with anomalous data points. As you can see, anomalous data points exhibit high transaction volume. As mentioned earlier, these anomalous transactions include fraudulent credit card testing, internal testing, and transactions related to application failures.

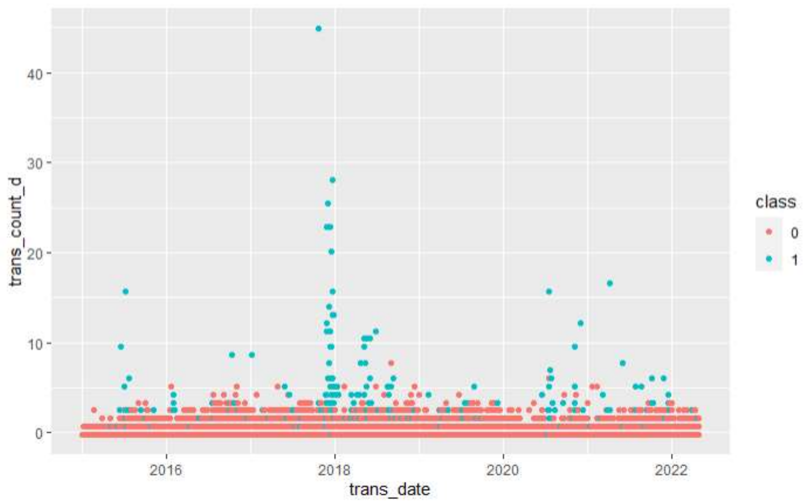
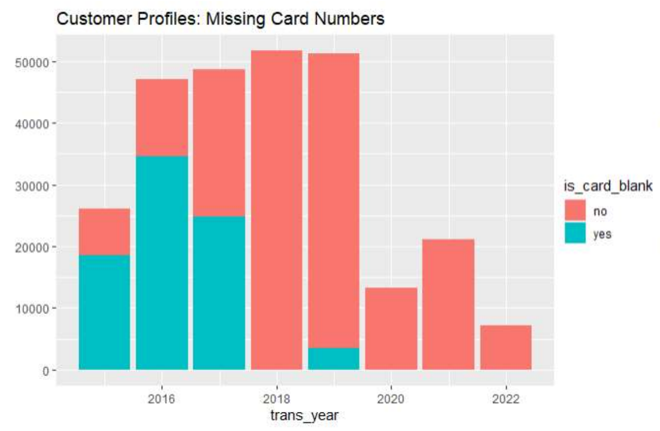


Fig 1. Number of daily transactions associated with anomalous data points

The raw dataset contains missing card (last 4 digits only) and customer IP address. Approximately 72% of the transactions pulled directly from the application database were missing card data and about 46% were missing customer IP information. Figures 2 and 3 below shows the distribution of missing data after incorporating card data from the payment gateway.



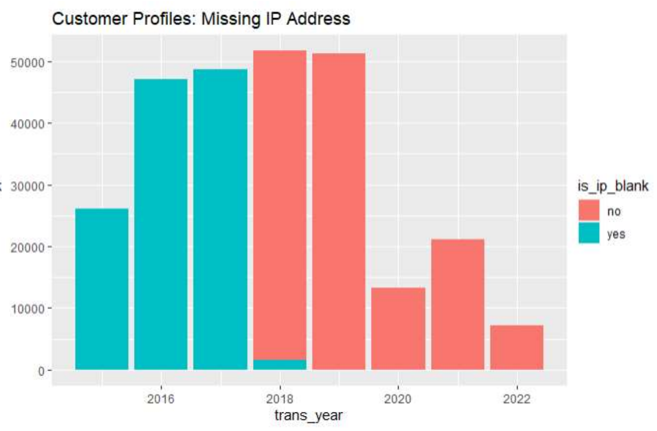


Fig. 2. Distribution of customers with no card data

Fig. 3. Distribution if customers with no IP address

To reduce the number of missing card data, archived and recent downloads from the card payment gateway, which logged the last four digits of the cards, were compiled and matched back against the raw dataset. After the match back process, the missingness was reduced to 28%. Analysis of the distinct customer card and IP address before imputation show that the mean number of different cards and IP addresses associated with a customer each year is 1.03 and 1.10, respectively. After imputing the missing data through *mice R* package, the mean changed to 1.02 and 1.08, which indicates that based on the pattern that the imputation algorithm determined, most customers have one card and are associated with one IP address each year.

Based on intuition and domain knowledge, certain metrics were calculated. This includes count-oriented measurements such as the number of daily and yearly transactions for a given customer, the number of distinct cards (by last four digits only) and IP addresses associated with a customer. Measurements related to amounts were also calculated such as the average transaction amount and the daily and yearly total amounts for a given customer. Ratio measurements were calculated such as the ratio of none-shippable transactions to the total transactions.

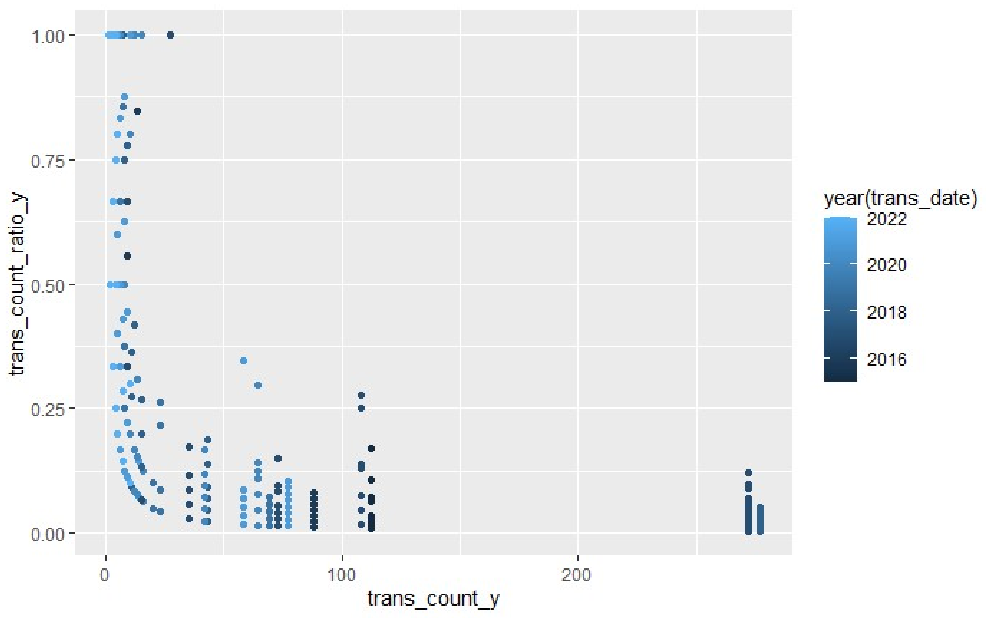


Fig. 4. No. of transactions for the year and ratio of day-to-year transactions.

As these metrics are made available, plotting some of these metrics gave some insights to the general patterns these anomalous transactions exhibit. One metric that is particularly interesting is the ratio between the number of transactions purchased so far during the day and the number of transactions purchased so far during the year. This ratio is one (1) for the first purchase during the year. This ratio decreases as the customer purchase more during the year. In the plot, you will see that a ratio of one tends to have a low transaction count, which is typical. However, there's a data point with a ratio of 1 and approximately 25 transactions. This data point represents 25 different purchases all made in one day. As the number of purchases during the year increase, this ratio decreases. Data points with low ratio and with relatively low transaction counts are the typical repeat customers who spread out their purchases throughout the year. As the ratio decreases and the transaction count increases, these are data points with high transaction volume and are spread out throughout the year. In the plot, we see customer profiles with transaction volumes of 50 to more than 300 transactions during the year.

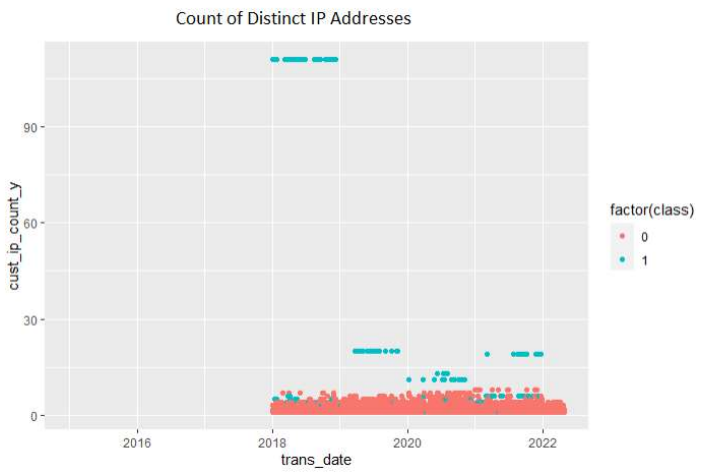


Fig. 5. Count of distinct IP addresses

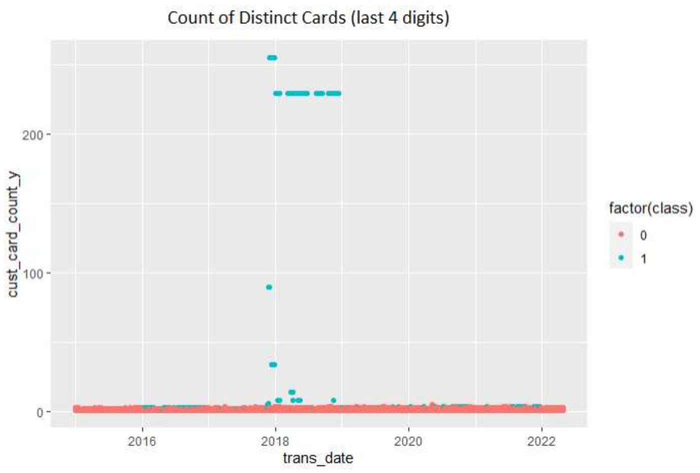
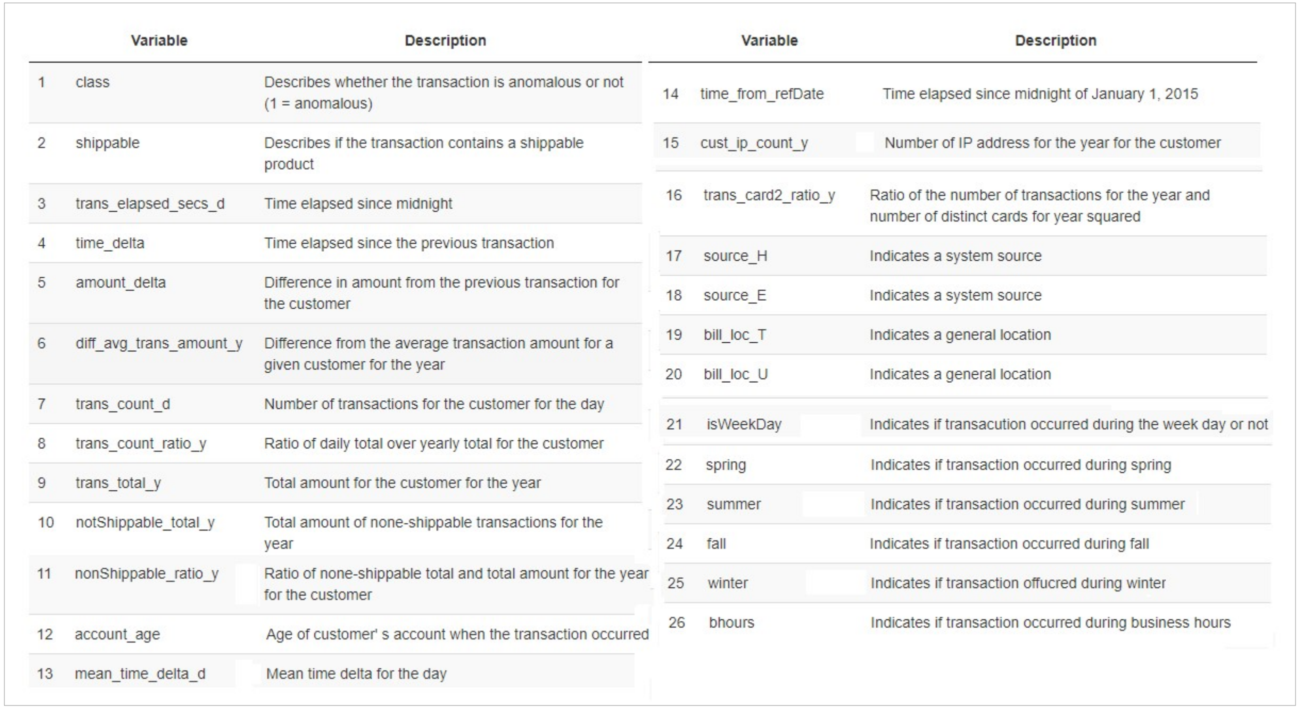


Fig. 6. Count of distinct cards by last 4 digits

An obviously interesting metric is the number of distinct cards and IP addresses associated with customer profiles. During the investigation phase, analysts noticed that these high-volume transactions were being processed with different credit cards and different IP addresses through a handful of accounts. Figures 5 and 6 above show that transactions associated with high card and IP counts are marked as anomalous in the dataset.

In addition, time-oriented metrics were calculated such as the amount of time elapsed since the previous transaction, the average time it takes for the next transaction to occur during the day, and the age of the customer account at the time of purchase. Also, categorical data such as whether the transaction occurred during business hours, the general location of the customer's billing address, and the season when the transaction occurred are included in the dataset.

The raw dataset contained fourteen variables. After performing the data transformations to include additional metrics, the number of predictor variables increased to thirty-two. Through the process of feature selection, which includes investigating near-zero variance variables and variables with collinearity problems, seven variables were dropped. The final dataset used for training and testing the models contained twenty-five predictor variables. Variables with units of measurements were scaled. Table 1 below outlines the final variables.



Variable		Description
1	class	Describes whether the transaction is anomalous or not (1 = anomalous)
2	shippable	Describes if the transaction contains a shippable product
3	trans_elapsed_secs_d	Time elapsed since midnight
4	time_delta	Time elapsed since the previous transaction
5	amount_delta	Difference in amount from the previous transaction for the customer
6	diff_avg_trans_amount_y	Difference from the average transaction amount for a given customer for the year
7	trans_count_d	Number of transactions for the customer for the day
8	trans_count_ratio_y	Ratio of daily total over yearly total for the customer
9	trans_total_y	Total amount for the customer for the year
10	notShippable_total_y	Total amount of none-shippable transactions for the year
11	nonShippable_ratio_y	Ratio of none-shippable total and total amount for the year for the customer
12	account_age	Age of customer' s account when the transaction occurred
13	mean_time_delta_d	Mean time delta for the day
14	time_from_refDate	Time elapsed since midnight of January 1, 2015
15	cust_ip_count_y	Number of IP address for the year for the customer
16	trans_card2_ratio_y	Ratio of the number of transactions for the year and number of distinct cards for year squared
17	source_H	Indicates a system source
18	source_E	Indicates a system source
19	bill_loc_T	Indicates a general location
20	bill_loc_U	Indicates a general location
21	isWeekDay	Indicates if transacution occurred during the week day or not
22	spring	Indicates if transaction occurred during spring
23	summer	Indicates if transaction occurred during summer
24	fall	Indicates if transaction occurred during fall
25	winter	Indicates if transaction offucred during winter
26	bhours	Indicates if transaction occurred during business hours

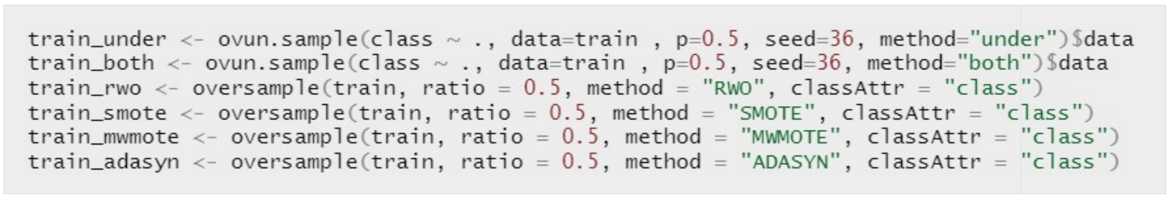
Table 1. Describes the final variables included in building the models

Part of the inherent nature of an anomalous behavior such as card fraud is that it is a rare event. As a result, datasets that describe anomalous activities such as fraud is highly imbalanced and skewed. Particularly, this project's dataset only has 0.75% data points in the minority class. The distribution of the dataset is significantly skewed. This imbalance needs to be addressed; otherwise, this can lead to a model that is biased towards the majority class. In this research, various sampling algorithms were considered to address this imbalance. A stratified random sampling of the majority class was performed to generate the majority data points that will be included in the training and testing dataset. About 74% of the majority class were selected from system source *E*, 22% from source *H*, and 4% from source *W*. The 2,291 anomalous transactions were then combined with the majority class to create the training and testing dataset. The training-testing dataset was then split into 80-to-20 ratio. To incorporate a much wider range of the different types of transactions that are in the majority class and to test how well the models perform with the various oversampling algorithms, the goal was to increase the size of the training-testing dataset as much as possible. However, due to lack of memory and computing power, the largest size that this study was able to process was 20,000 data points, of which 88.55% are in the majority class and 11.45% in the minority class. Among the different classifiers that were being trained, Random Forest presented a bottleneck with regards to the computing environment. Larger training datasets appear to take a very long time to complete. To give the algorithm time to train on approximately 30,000 data points, the process was left alone for about eight hours. However, after eight hours, the training was still not complete. There were other shorter time allotments attempted previously of about 2 to 3 hours. Those were also not successful. The processing was done on *R Studio* on a Windows 10 computer with 16 GB of memory and Corei7 Intel CPU.

		Sampling Algorithms Training Set Size					
Stratified Training-Testing Size	**Testing Size (80:20 split) 11.45% minority**	**Undersampling 50.01% minority**	**Combined Undersampling & Oversampling 50.17% minority**	**RWO 33.33% minority**	**SMOTE 33.33% minority**	**MWMOTE 33.33% minority**	**ADASYN 33.33% minority**
20,000	3,999	3,659	16,001	21,252	21,252	21,252	21,252

Table 2. Size of testing and training sets for each sampling algorithms

Table 2 above describes the testing size and the various training set sizes for each sampling algorithms used. Twenty thousand (20,000) data points were sampled from the entire dataset, and this was split into 80:20 ratio to form the training and testing sets, respectively. 11.45% of the testing set comprises of the minority class, which is significantly higher than the 0.75% distribution of the minority class in the entire dataset. Figure 7 below shows the code that generated the sampled training sets. The *ovun.sample* function of the *ROSE R* package was used to generate the undersampling and combined undersampling and oversampling training sets.



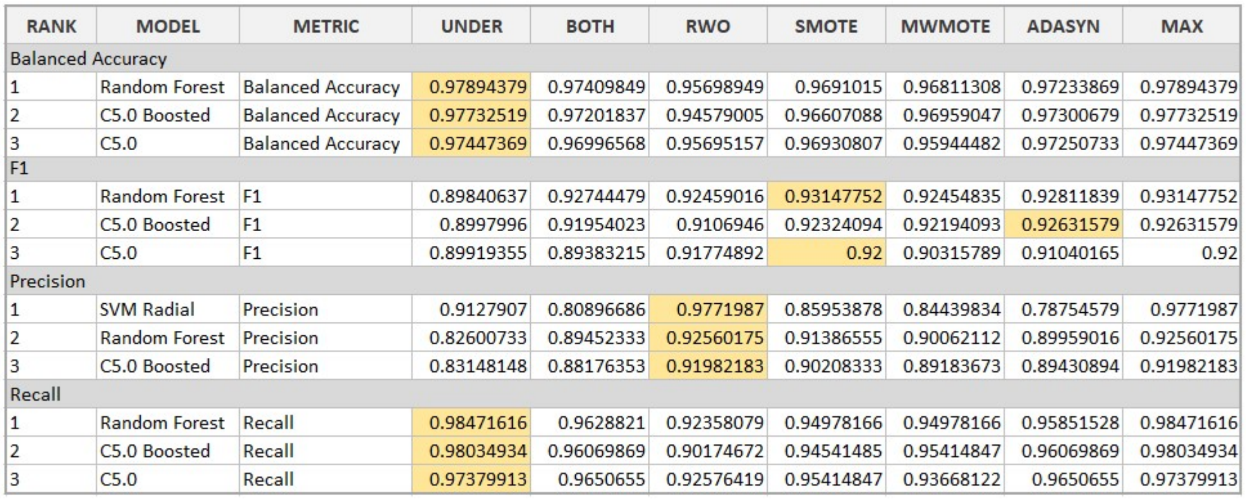
```
train_under <- ovun.sample(class ~ ., data=train , p=0.5, seed=36, method="under")$data
train_both <- ovun.sample(class ~ ., data=train , p=0.5, seed=36, method="both")$data
train_rwo <- oversample(train, ratio = 0.5, method = "RWO", classAttr = "class")
train_smote <- oversample(train, ratio = 0.5, method = "SMOTE", classAttr = "class")
train_mwmote <- oversample(train, ratio = 0.5, method = "MWMOTE", classAttr = "class")
train_adasyn <- oversample(train, ratio = 0.5, method = "ADASYN", classAttr = "class")
```

Fig 7. Code snippet to create the sampled training set

The wrapper function *oversample* of the *Imbalance R* package was used to generate the sampled training sets for RWO, SMOTE, MWMOTE, and ADASYN. As you can see, a ratio of 0.5 was specified; however, the actual distribution of minority and majority class in the respective training sets is 33.33%. The distribution generated by *ROSE* for the undersampling, and combined sampling is approximately 50%. This difference in distribution in the training sets may have an impact in the models. The findings described in the *Results* reflect the sampled training sets in figure 7 above.

Results

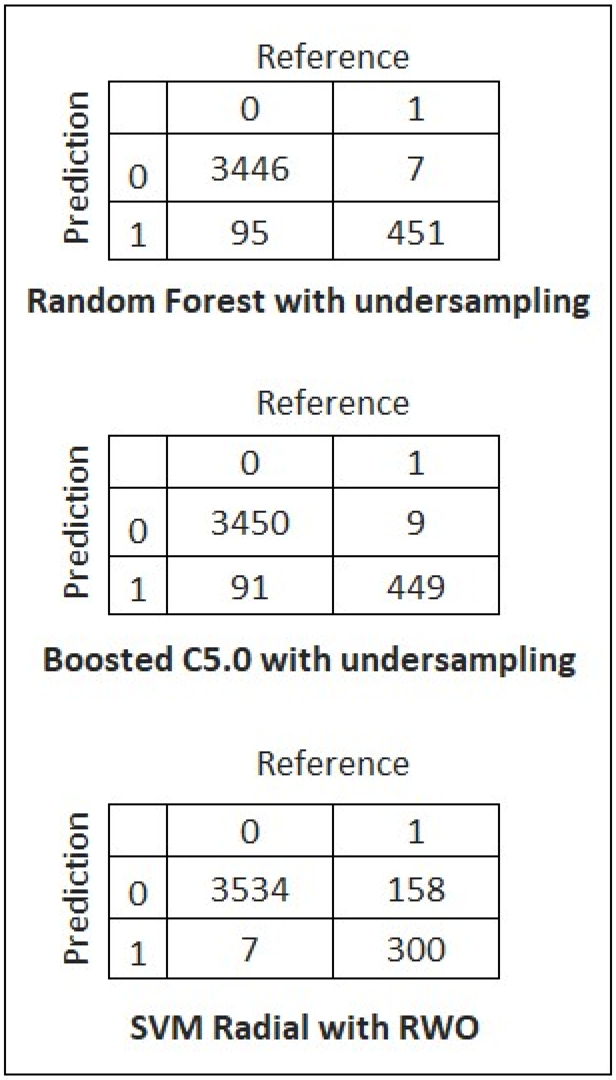
The metrics chosen to compare the models are recall, precision, F1, and accuracy. Each model is trained with six different sampling algorithms (undersampling, combined undersampling and oversampling, RWO, SMOTE, MWMOTE, and ADAYSN). For each metric, the highest score was selected across the different sampling algorithms. The winning model and sampling technique are then identified.



RANK	MODEL	METRIC	UNDER	BOTH	RWO	SMOTE	MWMOTE	ADASYN	MAX
Balanced Accuracy									
1	Random Forest	Balanced Accuracy	0.97894379	0.97409849	0.95698949	0.9691015	0.96811308	0.97233869	0.97894379
2	C5.0 Boosted	Balanced Accuracy	0.97732519	0.97201837	0.94579005	0.96607088	0.96959047	0.97300679	0.97732519
3	C5.0	Balanced Accuracy	0.97447369	0.96996568	0.95695157	0.96930807	0.95944482	0.97250733	0.97447369
F1									
1	Random Forest	F1	0.89840637	0.92744479	0.92459016	0.93147752	0.92454835	0.92811839	0.93147752
2	C5.0 Boosted	F1	0.8997996	0.91954023	0.9106946	0.92324094	0.92194093	0.92631579	0.92631579
3	C5.0	F1	0.89919355	0.89383215	0.91774892	0.92	0.90315789	0.91040165	0.92
Precision									
1	SVM Radial	Precision	0.9127907	0.80896686	0.9771987	0.85953878	0.84439834	0.78754579	0.9771987
2	Random Forest	Precision	0.82600733	0.89452333	0.92560175	0.91386555	0.90062112	0.89959016	0.92560175
3	C5.0 Boosted	Precision	0.83148148	0.88176353	0.91982183	0.90208333	0.89183673	0.89430894	0.91982183
Recall									
1	Random Forest	Recall	0.98471616	0.9628821	0.92358079	0.94978166	0.94978166	0.95851528	0.98471616
2	C5.0 Boosted	Recall	0.98034934	0.96069869	0.90174672	0.94541485	0.95414847	0.96069869	0.98034934
3	C5.0	Recall	0.97379913	0.9650655	0.92576419	0.95414847	0.93668122	0.9650655	0.97379913

Table 3. Top three performing models based on accuracy, F1, precision, and recall.

Table 3 above shows the top three models with the highest metric scores for each metric category. The scores highlighted are the winning scores under their respective sampling algorithms. For balanced accuracy, Random Forest scored the best, followed by Boosted C5.0 and C5.0. The sampling technique that performed the best with accuracy is undersampling. For F1 score, Random Forest performed the best with SMOTE, and again followed by Boosted C5.0 and C5.0. The sampling algorithms that performed the best with F1 are SMOTE and ADASYN. For precision, SVM RBF has the highest score, followed by Random Forest and Boosted C5.0. RWO sampling algorithm performed the with precision. Finally, for recall, Random Forest performed the best, followed by Boosted C5.0 and C5.0, and undersampling performed the best. Across all four metrics, Random Forest and Boosted C5.0 are consistently in the top three performers. Models that did not make it to the top three are CART and SVM linear. Table 4 below shows the confusion matrices of the top three models.



Reference

Prediction	0	1
0	3446	7
1	95	451

Random Forest with undersampling

Reference

Prediction	0	1
0	3450	9
1	91	449

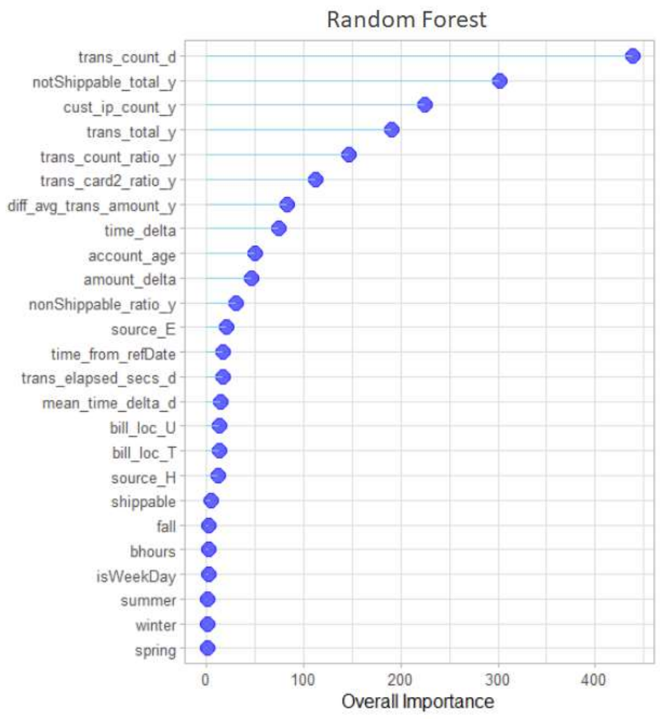
Boosted C5.0 with undersampling

Reference

Prediction	0	1
0	3534	158
1	7	300

SVM Radial with RWO

Table 4. Confusion Matrices of Random Forest, Boosted C5.0, and SVM radial



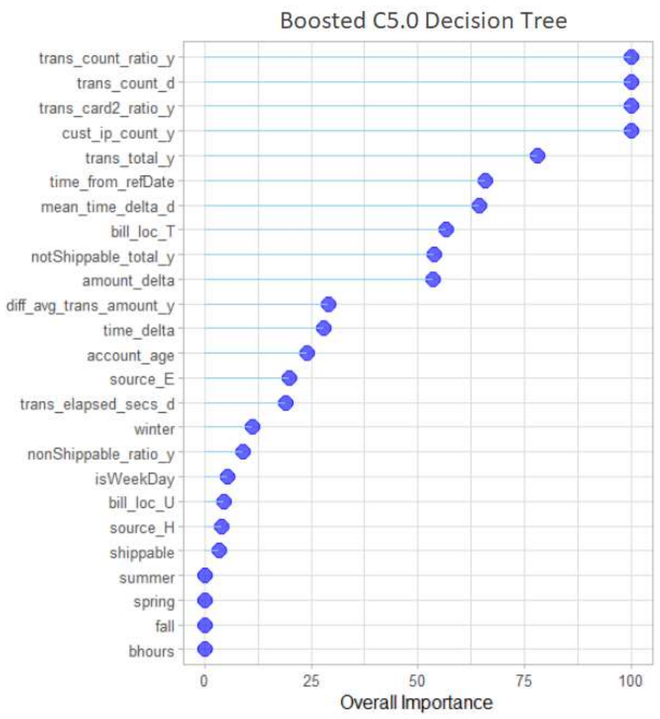


Table 5. Variable of Importance from Random Forest and Boosted C5.0 Decision Tree

Table 5 above shows the variable of importance plot from the Random Forest and Boosted C5.0 Decision Tree models. The top five variables for Random Forest are *trans_count_d*, *notShippable_total_y*, *cust_ip_count_y*, *trans_total_y*, and *trans_count_ratio_y*. These are, respectively, the number of transactions for the day, the total amount of none-shippable transactions for the year, the number of distinct customer IP addresses for the year, the total amount of transactions for the year, and the ratio between the number of transactions for the day and the number of transactions for the year. This research was curious about the significance of *trans_count_ratio_y* metric, and the variable of importance plot shows that this metric appears to be important. It's worth noting that another ratio-based metric, *trans_card2_ratio_y*, ranked number six for Random Forest. This metric was created to try and capture the relationship between the number of transactions and the number of distinct cards in such a way that would make a ratio of one meaningful to only one transaction and one card, which is the most common yearly pattern. This was achieved by squaring the number of cards in the ratio. With this metric available, the variable *cust_card_count_y*, which is the number of distinct cards for the year, was dropped since it contributed to the collinearity problem.

For Boosted C5.0 Decision Tree, the variables *trans_count_ratio_y* and *trans_card2_ratio_y* ranked first and second place, respectively, in terms of importance. Variables *trans_count_d*, *cust_ip_count*, and *trans_total_y* are also in the top five important variables as in Random Forest. Overall, both Random Forest and Boosted C5.0 Decision Tree models identified the same top five important variables.

Discussion

The results show that Random Forest ranked the best across all performance metrics except precision. The overall performance of Random Forest in this study is consistent with the results seen by the research done by Varmedja, et al., which concluded that Random Forest gives the best results in terms of classifying whether transactions are fraud or genuine. The study further states that Random Forest can give similar results as a simple neural network. In addition, their research mentions that high precision and recall were achieved after undersampling the data (Varmedja, et al.). In comparison, the results of this research show that undersampling performed the best for recall and accuracy across the top three models.

Boosted C5.0 also consistently ranked in the top three models across all the performance metrics. In the study by Sahin and Duman that compared various Decision Trees and SVM models, they found that C5.0 performed well with regards to accuracy, and they have chosen C5.0 and CART models as the final models to build their prediction model. However, in my research, CART did not rank in the top three for any of the performance metrics. In addition, their research also found that Decision Trees performed better than SVM with smaller training sets (Sahin and Duman). The findings of my research showed similar results. Boosted C5.0 and C5.0 performed better than SVM on all metrics except precision. SVM RBF performed the best for precision.

The model of choice is Random Forest. To achieve high recall and accuracy, undersampling would be the choice. It is interesting to note that with undersampling, the size of the training set is significantly smaller as described in table 2 when compared to synthetic oversampling techniques. As a result, the undersampled training set is much faster to train using computing and memory-intensive algorithms like Random Forest.

Conclusion

In summary, this research utilized real credit card dataset to train six classification models with Decision Trees, Random Forest, and SVM to predict anomalous transactions. Six sampling algorithms were considered: undersampling, a combination of both undersampling and oversampling, and various synthetic oversampling techniques (RWO, SMOTE, MWMOTE, ADASYN). Random Forest consistently ranked in the top three best classifiers across all performance metrics: accuracy, precision, recall, and F1. Random Forest is clearly a strong candidate for building classification models to identify anomalous transactions. Undersampling performed best with accuracy and recall. RWO sampling technique performed best with precision. SMOTE and ADASYN performed the best with F1. In addition, meaningful metrics were created and explored as potential features. The importance variable plots for Random Forest and Boosted C5.0 show that some of the metrics have predictive importance.

To answer the two questions posed earlier, the findings of this research showed that Random Forest outperformed the other models in all performance metrics except precision. It also showed that Decision Trees (C5.0 and Boosted C5.0) performed better than SVM across all performance metrics except precision.

For environments with low computing and memory power, undersampling, which has a significantly smaller training size, provide good performance with Random Forest and can be trained relatively fast. A classifier that aims to catch as many anomalous transactions as possible, a model that is recall-oriented would be a good choice; however, this needs to be balanced with an acceptable level of precision to maintain a manageable level of false positives. In this study, Random Forest and SVM RBF performed best on recall and precision, respectively.

Based on the findings of this research, the recommendation to the small business organization is to use Random Forest with undersampling for further future work since this combination provides the best recall at 98.47% and an acceptable precision at 82.60%.

This study only investigated one stratified sampling of the majority class from the entire dataset to form the training and testing sets. For future research, more samplings can be done to investigate how the models perform with different data points. In addition, future work can also investigate the performance of the classification models as the size of the training set is increased. It is worth noting that improved computing and memory power is required to accomplish this.

References

"Card Fraud Losses Reach $28.65 Billion." *Nilson Report – Card Fraud Losses Reach $28.65 Billion*, https://nilsonreport.com/mention/1313/1link/.

"Credit Card Transaction Fraud Continues to Climb to New Heights." *NCR*, 14 Mar. 2022, https://www.ncr.com/blogs/payments/credit-card-fraud-detection.

Kaggle. *Credit Card Fraud Detection*. https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

Lee, Nathaniel. "Credit Card Fraud Will Increase Due to the COVID Pandemic, Experts Warn." *CNBC*, CNBC, 1 Feb. 2021, https://www.cnbc.com/2021/01/27/credit-card-fraud-is-on-the-rise-due-to-covid-pandemic.html.

Mullen, Caitlin. "Card Industry Faces $400B in Fraud Losses over next Decade, Nilson Says." *Payments Dive*, 14 Dec. 2021, https://www.paymentsdive.com/news/card-industry-faces-400b-in-fraud-losses-over-next-decade-nilson-says/611521/.

Sahin, Y, and Duman, E. "Detecting Credit Card Fraud by Decision Trees and Support Vector Machines." Proceedings of the International MultiConference of Engineers and Computer Scientists 2011 Vol I, IMEC, March 16-18, 2011, Hong Kong.

Tinapunan, S. Data698. GitHub. https://github.com/Shetura36/Data698.

Varmedja, Dejan, et. al. "Credit Card Fraud Detection - Machine Learning Methods." 18th International Symposium INFOTEH-JAHORINA. March 2019, https://ieeexplore.ieee.org/document/8717766.