Improving Credit Card Fraud Detection by Profiling and Clustering Accounts

Navin Kasa, Andrew Dahbura, Charishma Ravoori, Stephen Adams
University of Virginia, nk4xf, amd6ua, cr2st, sca2c@virginia.edu

Abstract - **Credit card fraud is a problem that can cost banks billions of dollars annually, leading to increased incentives among financial institutions for the development of fast, effective and dynamic fraud detection systems. This research paper addresses credit card fraud detection through a semi-supervised approach, in which clusters of account profiles are created and used for modeling classifiers. Accounts are profiled based on their behavioral trends and clustered into similar groups. Groups are further identified as distinct customer segments based on purchase characteristics such as amount, frequency or distance. Random forest and XGBoost classifiers are trained on an entire sample and compared against classifiers trained at the transaction level across each cluster. This research concludes that the overall weighted performance of classifiers trained at the cluster level does not significantly outperform classifiers trained on the full sample. However, this research finds that clustering can be used to find meaningful groups of account holders that also have varying fraud rates across each cluster. Additionally, some classifiers trained on specific clusters yield significant improvements in performance over the baseline, whereas classifiers for other clusters do not perform as well as the baseline. This research also concludes that the optimal classifier for a given cluster varies by cluster, highlighting the potential for further development of new classifiers which may perform well on clusters that currently exhibit underperforming models.**

Index Terms – Fraud Detection, Machine Learning, Semi-Supervised Algorithms, Clustering

INTRODUCTION

The loss due to credit card fraud amounted to $22.8 billion in 2017 and was expected to rise to an amount of $31.8 billion by the end of 2018 [1]. M-commerce adoption has experienced a steep increase over the past few years, accompanied by the increase in credit card fraud in terms of both transaction and dollar volumes [2]. To sustain the growing adoption of M-commerce and other digital channels, fraud detection techniques that are efficient both in terms of cost and speed are needed. The need to maintain the balance between consumer experience and safety is one of the primary challenges for credit card fraud detection.

Failing to identify a fraudulent transaction will lead to financial loss and loss of consumer trust, which in turn could result in a decrease in future revenue. On the other hand, incorrectly tagging a valid transaction as fraudulent can hurt consumers' experience. Credit card fraud is constantly evolving with regards to speed and fraudster behavior , with fraudsters adopting techniques like BotNet transactions and synthetic identities [2]-[3]. The primary target of these methods is to make faster fraud transactions by taking advantage of digital banking platforms. The rapidly evolving nature of credit card fraud techniques demands quicker identification and deployment of fraud detection systems.

Supervised algorithms are proven to be efficient in fraud detection [4]-[5]. However, supervised techniques require labeled data, which requires organizations to continuously and accurately label each transaction as fraudulent and non-fraudulent. This process can be expensive and slow and the success of supervised techniques depends on the accuracy of labeled data. Another key concern with labeled data is the class imbalance which is unavoidable in this domain. This is because there are fewer fraudulent transactions than non-fraudulent transactions. Unsupervised algorithms are effective in identifying underlying patterns of unlabeled data. Credit card fraud transactions often highlight behavioral patterns of the cardholder [6]. Fraud detection techniques that can identify anomalies in a specific consumer's behavior could therefore be more effective. Hybrid methods bring together supervised and unsupervised techniques and are observed to be effective in handling the challenges previously outlined [7].

In this paper we discuss a hybrid approach of identifying groups of customers based on engineered behavior profiles and then building classifiers specific to those groups. Our approach has three steps: 1) profiling accounts 2) clustering the accounts and 3) building the classifiers. Multiple aggregated account level features are engineered from transactional level data in the original dataset.

Accounts are clustered into groups by the unsupervised K-means clustering algorithm. Clustering ensures that the accounts with feature values closer to each other are grouped together while separated from those that are dissimilar. Therefore, clusters should contain accounts that exhibit similar behaviors.

Supervised classifiers such as Random Forest and XGBoost are built on each cluster. Our hypothesis is that clustering improves the overall prediction capability of these classifiers. We test this hypothesis by comparing the Area

978-1-7281-0998-5/19/$31.00©2019 IEEE

Under the Curve (AUC) metric of classifiers trained on the entire dataset against the AUC for classifiers built on individual clusters.

This research will first provide a brief description of the data and the generation of account level features, followed by our approach which compares the results of classification before and after clustering.

RELATED WORK

There has been significant development in the area of credit card fraud in the past through supervised learning, unsupervised learning, and deep learning among others.

Supervised learning algorithms attempt to find a function that can effectively map the input to the corresponding output in labeled training data. Gabriel Rushin et al discuss an approach that compares the abilities of various supervised classification models such as Logistic Regression, Gradient Boosted Trees (GBT), and Neural Networks in detecting fraud [4]. They also explore feature generation using both domain expertise and an autoencoder – an unsupervised method for feature engineering. Using these two methods along with the original dataset, they create six feature sets on which to test the classifiers. The results of their study show that the feature set that was created using domain expertise managed to raise the AUC value by 1-4% while the autoencoder features added no improvement. Amongst the supervised classifiers, Neural Networks performed the best on a majority of the datasets, followed by GBT. Although GBT showed adequate predicting powers, Extreme Gradient Boosting Trees (XGBoost) has shown to be more effective in the credit card fraud domain as seen in the work by Sahil Dhankhad et al [8].

The comparative study by Sahil Dhankhad et al. further analyzes supervised methods that classify credit card fraud and their performance against a custom super ensemble method. They deduce that the best performing (according to precision and recall) classifiers are Random Forest, XGBoost, and the custom ensemble method. The worst performing classifiers in their study are Decision Tree, K-Nearest Neighbors (K-NN), and Naïve Bayes. Since credit card fraud data is generally highly imbalanced, the study employs an under-sampling technique to counter the biasing effects imbalanced data has on classification.

Another supervised research methodology can be seen in the paper by Abhimanyu Roy et el [5]. The study makes use of domain expertise engineered features and compares the performance of different kinds of neural networks like, Artificial Neural Networks and Recurrent Neural Networks among others. Ultimately, the Gated Recurrent Unit out-performed the other methods. This study demonstrated the predictive power of deep learning methods that utilize time-series information.

Unsupervised machine learning is a class of algorithms that attempt to find patterns in the data with no previously labeled training data. One of the most commonly used unsupervised methods is clustering. Generally, K-means clustering is used on credit card data, as seen in the work by D. Viji et al [9] and Vaishali [10]. In the research conducted by Vaishali, K-means clustering is applied to randomly generated data to split it into groups based on how likely a fraudulent transaction is to occur. To account for the non-numeric attributes, One-Hot-Encoding (among other methods) must be applied. While K-means and K-modes clustering only work with numeric and categorical attributes respectively, K-prototypes, an alternative, is a hybrid method that applies characteristics of both to the data. This aspect makes K-prototypes useful since credit card data tends to have a mix of numeric and categorical attributes. The algorithm, as described by Zhexue Huang [11], works by assigning each data point to a cluster with the prototype closest to it as shown by the similarity measure. The method dynamically updates the K-prototypes, in each iteration, to maximize the similarity of the data points within a cluster and the dissimilarity of the data points in different clusters.

A completely different approach to solving this problem can be seen in the study conducted by Adrian Mead et al [12] in adversarial learning using a reinforcement method called the Markov Decision Process. A reinforcement approach uses signals and rewards to train a system, an agent interacting with its environment, rather than labels or clusters. Whenever the agent makes a decision or changes its state, the environment processes the change and returns some feedback. The main purpose of the agent in this scenario is to maximize the positive feedback or cumulative reward. In the study, the agent was a fraudster and the environment was the bank's fraud classifier, for which a logistic regression classifier was used. This adversarial framework addresses the changing behavior of fraudsters in an attempt for banks to develop dynamic fraud detection systems and classifiers.

DATA

The dataset is provided by a bank and contains approximately 80 million transactions across 1.1 million account holders. These transactions were recorded over an eight-month period and have 69 features. The dataset captures various characteristics of the customer spending patterns such as time, amount, location, type of point of sale, currency etc. The dataset also captures the contextual and operational information of a transaction like the safety capabilities of point of sale, type of authorization request, presence of account holder at point of sale etc. Account specific features like number of credit cards and account product code are also available.

The key challenge with this dataset is the significant class imbalance. Only 0.136% of the total transactions belong to the fraud class. Additionally, only 26,000 accounts have fraud transactions associated with them. Hence to capture the behavioral patterns associated with fraud at the account level, it is necessary to study these minority class accounts in detail. Multiple sample datasets are generated from the full dataset for deeper examination.

I. Data Preprocessing

The data contains numerical and categorical features, both of which contain missing values. All features containing missing values for more than 80% of the transactions were removed to avoid synthetic data influencing the results, if imputed. All remaining missing data is imputed with means for numerical features and mode for categorical features. No transactions were removed in this process.

II. Sampling

Five samples were generated from the original dataset for this study. Each sample contains close to 110,000 accounts and 5 million transactions. All fraud accounts are included in each sample yielding close to 2% fraud rate. The rest of the 84,000 accounts are randomly picked from accounts containing no fraud transactions. Non-fraud accounts were sampled without replacement. No oversampling of the minority class is implemented.

APPROACH

Conventional supervised approaches to credit card fraud build classifiers on transaction data across all accounts within a bank's customer dataset. Features are engineered at the transaction level irrespective of the origin account, and used to predict fraud. This research study utilizes a more complex approach, in which features are first engineered using account level data rather than transaction level data. These accounts are then clustered into distinct groups based on behavioral patterns, which serves helpful in identifying unique customer groups that were previously unknown or not captured. Classifiers are then built on each cluster of accounts using their respective sets of transactions to train the classifier.

I. Account Profiling

Feature engineering that can capture the rate of change over time and variance of various transactional features has proven to be effective in credit card fraud detection domain [4]-[5]. Previous work focused on profiling transactions to identify transaction level trends. We developed an analytical framework to profile accounts and identify behavioral trends at the account level on four dimensions – Spend, Spread, Safety and Sketch. Multiple aggregated features are built for each dimension. A detailed description of these dimensions is provided below:

- **Spend**: Spending patterns are measured under this dimension. To achieve this the mean and variance of the data associated with an account's transactions are calculated. This data includes the daily spending rate and the time between transactions.
- **Spread**: Diversity of spending is measured under this dimension. To achieve this the geographical spread and types of merchants handling the transactions are observed. This data includes the daily merchant count, distance of point-of-sale from home and merchant categories.
- **Safety**: The safety preferences of an account are measured under this dimension. To achieve this the preferred channel of transaction and safety capabilities of point-of-sale are observed.
- **Sketch**: The characteristics of an account irrespective of transactions are measured under this dimension. This data includes most preferred account type on which transactions are recorded and mean and variance of money in account before each transaction are computed.

II. Clustering

Accounts are clustered using the engineered account profiles. Account level profiles include both numeric and categorical features. Categorical features are one hot encoded to binary numeric form and numerical features are scaled using the min max scaler in order to give equal importance to all features. The min max scaling approach shrinks the range of numeric features such that they are between zero and one. The min max scaler works well in the case where the data inputs are a combination of continuous and binary features, however unit normal scaling or alternative scaling methods are also appropriate.

K-means is an unsupervised method used to separate unlabeled data into K distinct subgroups based on a series of input features. The within cluster variation is the amount that observations within the same cluster differ from one another according to a specified comparison metric. A good clustering assignment will be one that minimizes the sum of the within-cluster variation across all clusters [12]. Euclidean distance is used as the metric for clustering the accounts. One challenge with K means clustering is determining the optimal number of clusters to use, especially in the case where the user lacks a deep understanding of the problem's domain or the goal is to identify new subgroups within the data. The sum of the within cluster variations at each level of clusters is plotted. In the ideal case, the sum of within cluster variations (in this case, sum of squared distances) is drastically reduced up to some number of clusters "K", whereon it falls insignificantly afterwards. In that case, the number of clusters that is appropriate for the dataset is set to "K".

III. Classification Methods

Random Forest and XGBoost classifiers are built on both sample level data and clusters. Categorical features are encoded into their numerical equivalents by target encoding to speed up building of classifiers. Since the current fraud rate is close to 2%, appropriate class weights are provided to both algorithms to adjust the impact of imbalance. For both models, 70 trees were fit, each allowing for a max depth of 6 splits and each split could consider 50% of the full feature set.

- Random Forest is an ensemble method that builds a 'forest' of decision trees with some measure of randomness introduced because each tree only selects the best feature for each split from a sample of all the

potential features. The classifier's final result is an aggregation (mode) of all the outputs of the individual trees [13]. Random forest automatically mitigates the problem of overfitting shown by decision trees since the output is not dependent on only one tree. It also provides an easy way to measure the relative importance of the features in the dataset. This is due to the nature of each decision tree, which splits the data on the attributes in the order of greatest contribution for classification.

- Extreme Gradient Boosting, known as XGBoost, is another tree-based learning algorithm. Boosting involves growing trees sequentially, where decision trees are fit to the residuals of the current model and then added into the model to update the residual values [11]. In this method, trees that are built will be dependent on previously grown trees. Boosting is also a slower learning algorithm which may fit many small trees, both of which can help prevent against potential overfitting [11]. A parameter grid was used to iterate over a range of potential parameter values to determine the optimal inputs for training the model. L1 regularization is also implemented.

RESULTS

After calculating the sum of the within cluster variations at each number of clusters from five through thirty, the appropriate number of clusters to use was determined to be 10. The sum of squared distances for clustering fell at a much steeper rate until reaching 10 clusters. Beyond 10 clusters the sum of squared distances fell much slower and a noticeable increase in training time occurred, making the tradeoff impractical for beyond 10 clusters.

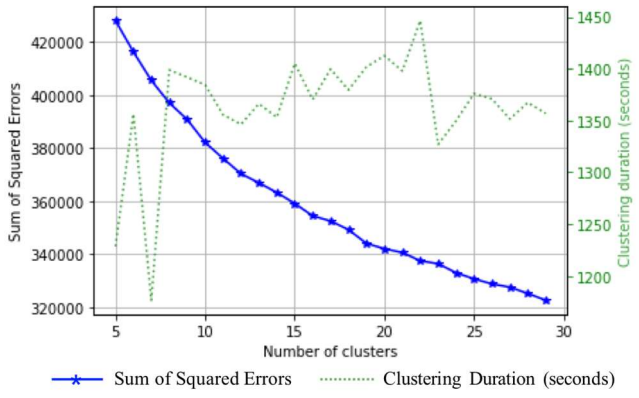

FIGURE I
SUM OF SQUARED ERRORS BY NUMBER OF CLUSTERS

After investigating the accounts in each cluster, specific customer groups were identified. These customer groups should be investigated further to understand where the cluster assignments or feature set may be improved by the bank. This will allow model performance to hopefully improve for underperforming clusters. Table 1 shows the proportion of accounts in each customer group across samples.

TABLE I
DISTRIBUTION OF ACCOUNTS BY CUSTOMER GROUPS

Customer Group	Proportion
Global Spenders	9.28%
Diverse Spenders	14.7%
Low Spenders	9.63%
Frequent Spenders	11.57%
Big Spenders	6.16%
Loyal Spenders	10.68%
General	14.22%
Local Spenders	13.02%
Big and Infrequent Spenders	3.65%
Glocal Spenders	7.08%

- **Global Spenders**: Customers who spend farther away from home
- **Diverse Spenders**: Customers who spend across a larger number of industries
- **Low Spenders**: Customers whose transactions are a lower dollar amount
- **Frequent Spenders**: Customers who have a higher frequency of transactions within a given time period.
- **Big Spenders**: Customers whose transactions are higher dollar amounts.
- **Loyal Spenders**: Customers with transactions concentrated over a smaller number of industries.
- **General**: No specifically discernible behaviors.
- **Local Spenders**: Customers who spend closer to home.
- **Big and Infrequent Spenders**: Customers whose transactions are higher dollar amounts and have a low frequency of transactions within a given time period.
- **Glocal Spenders**: Customers who regularly spend both close to and far away from home.

While the overall AUC does not improve, there is a significant change in the fraud rates observed across clusters within every sample, and some clusters with a higher fraud rate also notice a drastic improvement in model AUC performance compared to the base model. Table II shows the distribution of fraud in each cluster for a given sample.

TABLE II
FRAUD RATES ACROSS CUSTOMER GROUPS

Customer Groups	Average Fraud Rate	Standard Deviation
Global Spenders	6.36%	0.79%
Diverse Spenders	1.21%	0.11%
Low Spenders	2.17%	0.80%
Frequent Spenders	1.64%	0.33%
Big Spenders	4.74%	0.08%
Loyal Spenders	2.20%	0.74%
General	1.54%	0.30%
Local Spenders	1.34%	0.05%
Big and Infrequent Spenders	2.95%	0.08%
Glocal Spenders	2.93%	1.10%

After training Random Forest and XGBoost classifiers for each cluster, a weighted average AUC metric was calculated for clustering to compare model performance against the baseline model. The cluster weighted AUC was calculated by multiplying each cluster's AUC by the proportion of transactions in that cluster, and then combining each weighted AUC for a final system AUC for clustering. After clustering and considering for the total number of transactions in each cluster, there was no discernible change in the AUC value compared to the baseline performance.

TABLE III
AUC VALUES FOR BASELINE AND CLUSTER AVERAGES

Label						
	Sample1	Sample 2	Sample 3	Sample 4	Sample 5	Average
Baseline RF	0.848	0.848	0.847	0.849	0.849	0.848
Baseline XGBoost	0.857	0.858	0.857	0.857	0.858	0.857
Weighted Cluster AUC Average	0.856	0.858	0.854	0.856	0.856	0.856

For some clusters such as Big Spenders and Glocal the baseline with an increase in performance of approximately 0.03. However, for other clusters such as loyal spenders, the model's performance shows a drop by approximately 0.01. To address the change in performance across clusters, each cluster was investigated to determine whether its average value for any given feature(s) was drastically different from the average for the rest of the clusters.

the account level, the hypothesis is that clustering will be able to separate accounts into meaningful clusters that will improve prediction capabilities. Two baseline models without clustering were generated for comparison against cluster specific models. XGBoost achieved a higher AUC average across samples than random forest, while the weighted AUC after clustering remained unchanged.

However, after clustering there is a discernible difference in the fraud rates observed across clusters. For some clusters with higher fraud rates, the cluster specific classifiers are also outperforming the base model by as much as 0.03 AUC.

However, some clusters with marginally lower performance than the base model contain a large amount of transactions, which brings down the overall weighted performance. Further investigation is required to determine the reason that some clusters perform worse than the baseline model, and whether reallocating accounts in those underperforming clusters would result in better overall system performance.

Clustering was also able to determine distinct behavioral patterns across account holders for each cluster. These cluster behaviors also hold across samples. Knowing the customer groups and behavioral tendencies of clusters that perform well and those that perform poorly is valuable in trying to improve fraud detection.

TABLE IV
AUC VALUES FOR CUSTOMER GROUPS

Label	Random Forest					XGBoost					Average RF	Average XGBoost
	Sample1	Sample 2	Sample 3	Sample 4	Sample 5	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5		
Global Spenders	0.853	0.865	0.86	0.863	0.865	0.875	0.885	0.882	0.879	0.882	0.821	0.881
Diverse Spenders	0.85	0.876	0.849	0.834	0.867	0.845	0.865	0.845	0.799	0.86	0.855	0.843
Low Spenders	0.867	0.841	0.848	0.843	0.875	0.847	0.825	0.846	0.849	0.886	0.855	0.851
Frequent Spenders	0.859	0.853	0.85	0.862	0.85	0.848	0.848	0.824	0.855	0.849	0.855	0.845
Big Spenders	0.875	0.876	0.876	0.876	0.876	0.887	0.888	0.886	0.888	0.889	0.876	0.888
Loyal Spenders	0.843	0.866	0.859	0.839	0.839	0.831	0.868	0.864	0.824	0.828	0.849	0.843
General	0.86	0.843	0.817	0.823	0.827	0.852	0.839	0.809	0.809	0.805	0.834	0.823
Local Spenders	0.847	0.835	0.864	0.874	0.847	0.839	0.80	0.86	0.863	0.84	0.853	0.840
Big and Infrequent Spenders	0.872	0.867	0.873	0.87	0.871	0.874	0.87	0.877	0.873	0.875	0.754	0.874
Glocal Spenders	0.873	0.878	0.875	0.858	0.85	0.886	0.877	0.864	0.864	0.839	0.867	0.871

As seen in Table IV he Big Spenders customer group is experiencing an AUC improvement of 0.02-0.03 which will have a considerable reduction of losses due to fraud on a dollar basis. Customer groups with higher geographical spread of spending are also experiencing 0 .01-0.02 improvement in AUC after clustering. Since the Global Spenders customer group has the highest fraud rate, this improvement helps in preventing higher instances of fraud. The customer groups without any tangible patterns based on the engineering profiles have a drop in AUC which might indicate the need for further clustering of these groups to find behavioral patterns relevant for these accounts.

If model performance is best for groups such as Big Spenders, the model is likely to perform well on higher value fraud transactions which is valuable to the bank. Also, if one can determine which groups it is performing poorly on, it can work to engineer new features in those domains that may be more useful in detecting fraud and improving performance.

The secondary question of this research investigates whether different classifiers result in greater model performance depending on which cluster is evaluated. While the baseline XGBoost model performed better than random forest across all samples, this is not the case when studied at the cluster level. Specifically, random forest outperforms XGBoost across some clusters while XGBoost outperforms random forest for other clusters. This suggests promising results that training other classifiers such as logistic regression or neural networks may increase the overall weighted clustering AUC to a threshold that is statistically significantly greater than the baseline classifier achieves. In the future this research could be further developed by

CONCLUSION

The primary question of this research investigates whether clustering helps improve the predictive performance of credit card fraud. By engineering useful and descriptive features at

creating additional classifiers such as logistic regression and neural networks to test on underperforming clusters for various customer groups. Current results highlight the potential for optimal classifiers to vary by cluster, suggesting that these classifiers may boost overall fraud detection performance when evaluated using clustering. Additionally, account and transaction characteristics of each cluster should be investigated further to help understand what features are useful in dividing customers Specifically, clusters that cannot be differentiated between must be investigated further to better understand their customer behaviors. If banks can understand groups of consumers where models perform well or do not perform as well, they can begin to investigate and engineer new features that may be more useful to fraud models than the existing features.

Lastly, further research could investigate whether reassigning accounts in underperforming clusters to new clusters based on which cluster their feature values align closest to helps improve performance. It is possible that accounts on the fringe of two customer groups share characteristics that may be useful in predicting fraud when looked at jointly, but are missed by the current model. This also presents the task of determining when accounts should be assigned to new clusters as their behavioral patterns change over time.

ACKNOWLEDGEMENT

We would like to thank Capital One for insight and guidance on the use case for this paper. We are grateful to the University of Virginia advisors who validated results. Additionally, we thank David Lutz who provided guidance in numerous aspects of this project.

REFERENCES

[1] Nilson Report Issue 1118. 2017. Available:https://nilsonreport.com/upload/content_promo/The_Nilson_Report_Issue_1118.pdf. Accessed: 10th September 2018

[2] Lexis Nexis Risk Solutions 2018 True Cost of Fraud Study Available:https://risk.lexisnexis.com/insights-esources/research/2018-true-cost-of-fraud-study-for-the-retail-sector Accessed: 25th March 2019

[3] Zeager, Mary F. and Sridhar, Aksheetha et al. April 2017, "Adversarial Learning in Credit Card Fraud Detection." *2017 Systems and Information Engineering Design Symposium (SIEDS)*

[4] Rushin, Gabriel and Stancil, Cody et al. April 2017, "Horse Race Analysis in Credit Card Fraud—Deep Learning, Logistic Regression, and Gradient Boosted Tree Gabriel." *2017 Systems and Information Engineering Design Symposium (SIEDS)*

[5] Roy, Abhimanyu and Sun, Jingyi et al. April 2018, "Deep Learning Detecting Fraud in Credit Card Transactions." *2018 Systems and Information Engineering Design Symposium (SIEDS)*

[6] Bhattacharyya, Siddhartha Jha, Sanjeev et al. February 2011, "Data mining for credit card fraud: A comparative study". *Journal: Decision Support*

[7] Jiang, Changjun and Song, Jiahui et al. 15 March 2018, "Credit Card Fraud Detection: A Novel Approach Using Aggregation Strategy and Feedback Mechanism." *IEEE Internet of Things Journal*

[8] Dhankhad, Sahil and Mohammed, Emad et al. 2018, "Supervised Machine Learning Algorithms for Credit Card Fraudulent Transaction Detection: A Comparative Study", *IEEE International Conference on Information Reuse and Integration*

[9] D. Viji, and S. Kothbul Zeenath Banu, "An Improved Credit Card Fraud Detection Using K-Means Clustering Algorithm." *International Journal of Engineering Science Invention (IJESI)*

[10] Vaishali, July 2014, "Fraud Detection in Credit Card by Clustering Approach." *International Journal of Computer Applications* (0975 – 8887) Volume 98– No.3

[11] Huang, Zhexue, 1997, "Clustering large data sets with mixed numeric and categorical values." *Proceedings of the First Pacific Asia Knowledge Discovery and Data Mining Conference*, pp. 21-34

[12] Mead, Adrian and Lewris, Tyler et al. 2018, "Detecting Fraud in Adversarial Environments: A Reinforcement Learning Approach." *2018 Systems and Information Engineering Design Symposium (SIEDS)*

[13] James, Gareth and Witten, Daniela et al. 2013, "An introduction to statistical learning: with applications in R" New York: Springer.

AUTHOR INFORMATION

Navin Kasa M.S Student, Data Science Institute, University of Virginia.

Andrew Dahbura M.S Student, Data Science Institute, University of Virginia.

Charishma Ravoori M.S Student, Data Science Institute, University of Virginia.

Stephen Adams Senior Scientist, Systems and Information Engineering, University of Virginia.