Automating Fraud Analytics Common Point of Compromise Analysis

Western Governors University

Table of Contents

A. Project Overview

A1. Research Question or Organizational Need

This project will take a routine, manual, and time-consuming fraud analytics process known as a Common Point of Compromise/Purchase analysis and will create a Python application to automate this analysis. This analytic process is integral to the quality and effectiveness of fraud prevention control development.

A2. Context and Background

Fraud analytics is a fast-paced ecosystem where the adage ‘Time is money’ could not be more applicable. The time it takes for an organization to identify and respond to fraud events is positively correlated with the amount of money fraudsters can exfiltrate from the organization. One of the more time-intensive tasks occurring in fraud centers around the world is CPC/CPP or Common Point of Compromise/Common Point of Purchase analysis. This process involves the comparison of transaction histories from multiple payment cards where all cards have similar patterns of fraudulent activity. The goal of this analysis is to determine where card information was skimmed. Once a compromised merchant and date range are detected, targeted fraud controls can be implemented to mitigate future losses.

A3. Summary of Published Works

Breaches and data compromises at merchants around the globe have increased at an alarming rate in the last decade. Some compromises make the national news, such as Target in 2013, or Home Depot in 2014 and 2020. However, most of these compromises happen at a much smaller scale, such as a skimming device on a gas pump or ATM. Regardless of the source of the breach, countless hours go into the analysis and detection every day. Successful completion of a CPP analysis can lead to more effective controls for future fraud mitigation. A fraud control that can be focused on a small subset of a card portfolio will perform better than one that applies to more cards. Many financial institutions have hundreds of thousands, if not millions, of cards in their portfolios. Fraud controls allow these institutions to mitigate the losses associated with these compromises, rather than necessitate the issuance of physical cards, many of which may never be used fraudulently.

On March 5, 2014, in an article titled “Sally Beauty Hit By Credit Card Breach,” Brian Krebs detailed the use of CPP to mitigate fraud. Krebs, a renowned cybersecurity reporter, who has posted articles on many of the lesser-known data breaches in the last few years, summed up the CPP process eloquently when he wrote:

> On March 2, a fresh batch of 282,000 stolen credit and debit cards went on sale in a popular underground crime store. Three different banks contacted by KrebsOnSecurity made targeted purchases from this store, buying back cards they had previously issued to customers.

> The banks each then sought to determine whether all of the cards they bought had been used at the same merchant over the same time period. This test, known as “common point of purchase” or CPP, is the core means by which financial institutions determine the source of a card breach.
>
> Each bank independently reported that all of the cards (15 in total) had been used within the last ten days at Sally Beauty locations across the United States. Denton, Texas-based Sally Beauty maintains some 2,600 stores, and the company has stores in every U.S. state. (2014)

In this article, Krebs illustrates the process that most banks use in order to stay ahead of the constant wave of fraud they must face on a daily basis. The sheer quantity of the number of CPP analysis that a fraud department must conduct has led large analytics firms to create solutions touting the latest in machine learning and AI. These solutions, however, remain out of reach for the majority of organizations.

One such solution is outlined by this SAS whitepaper. SAS is one of the world’s leading analytic solutions providers, offering services to help large banks detect these CPPs. SAS’s solution uses a combination of AI and machine learning and ranks every transaction in a card’s history to gauge the potential risk at each merchant. It is quite an intriguing solution if the organization can also stomach the impressively large price tag. As Ian Holmes, Global Security Intelligence Practice at SAS writes:

> A crucial part of controlling card fraud is being proactive rather than reactive. For issuing banks, a critical aspect of that effort is the common point of purchase (CPP) analysis. CPP analysis identifies the likely merchant location from where card numbers were stolen so that banks can mitigate future fraud on other compromised cards. And since identifying a CPP requires a loss from fraud, the more automated – and therefore quicker – the identification is, the more fraud can be prevented. (n.d.)

RippleShot, a Chicago based breach detection company, is also one of the leading providers of merchant compromise detection. RippleShot uses a unique approach in which they use the combined knowledge of their entire customer base to help each organization detect compromises. Because RippleShot has access to a larger card base and transaction history than any one organization, they can help even the smallest organizations detect compromises. They state in their whitepaper that their solution is “4x faster than CAMS alerts” and “Reduces annual fraud losses by 15-25%”. While this is an impressive solution, that additional knowledge also comes with a hefty price tag.

The small regional financial institution where I worked has piloted both SAS’s and RippleShot’s CPP products. While these products were significantly quicker than our current manual process and almost as effective, they were prohibitively expensive. The high cost of these solutions leaves smaller banks and credit unions with no cost-effective option but to continue their manual and time-consuming analytics process. The reliance on time-consuming processes leads to ever-increasing losses. There is thereby a need, and market for, a solution to this ever-growing problem that can be done without the same expenditure.

A4. Summary of Data Analytics Solution

My solution to this extremely costly fraud analytics process will be to create a standalone Python application to assist analysts around the globe in quickly and easily finding these compromises. This will provide more time for analysts to complete other beneficial analysis and fraud mitigation practices. This application will require the user to input three or more cards. The application will then complete the common point of purchase analysis. The application will output a merchant name, a date range, and a list of additional card numbers used at that merchant during the compromise window. Expediting this analysis will allow the analyst to spend more time crafting rules and strategies for additional fraud mitigation.

A5. Benefit to Organization and Decision-Making Process

The two primary benefits to the organization are saving time and money. The time the organization will save can be measured in increased employee output. The fraud control decision-making process in the fraud department will be streamlined by this application. The analysts using the solution will reduce the time they need to complete the analytic decision-making process by eliminating the manual analysis needed to determine key elements of the compromise. The time previously spent on this manual analysis can then be reallocated to other pressing fraud mitigation tasks. In addition to time, the organization will save money through the increased expediency by which fraud is mitigated, this will result in the reduction of fraud losses.

B. Data Analytics Plan

B1. Goals, Objectives, and Deliverables

The goal of this project is to create a Python application to automate the fraud analytics Common Point of Purchase/Compromise analysis.

The objectives for this goal are:

- Determine which merchant is the source of compromise.
    - The deliverable for this objective is for the application to return the name of the compromised merchant.
- Determine the date range of the compromise.
    - The deliverable for this objective is for the application to return the date range when the merchant was compromised.
- Determine other payment cards that may be at risk of being compromised.
    - The deliverable for this objective is for the application to return a list of cards that were used at the compromised merchant during the compromise date range.

B2. Scope of Project

The scope of this project will include a standalone Python application. This application will use analyst provided card numbers as its input. The output of this application will be compromised merchant name, compromise date range, and additional cards used at the compromised merchant during the compromise date range. The scope of this project will not include the detection of multiple merchant compromises, or network level breaches. Also, the scope of this project will not include card not present or online merchant compromises.

B3. Standard Methodology

This project will utilize a Waterfall project methodology. This methodology consists of five major steps where the first must be completed before the next can be undertaken. Those five steps include Requirements, Design, Implementation, Verification, and Maintenance. The following explains how I plan to proceed during each of these phases.

Requirements: One of the key aspects of the Waterfall methodology is that all customer requirements are gathered initially. In this step, I will determine the project scope, the user expectations, and the resources needed to complete the project.

Design: In this stage of the methodology, I will compile the tasks needed to be completed to achieve the project objectives. Some of these tasks include determining the type of analytical method needed to detect a compromised merchant programmatically, the steps needed to determine the date range of compromise, how the information will be gathered by the program, and how the data will be output.

Implementation: In this stage, I will complete the tasks needed to achieve the objectives and test the application to ensure it is producing the desired results.

Verification: In this stage, I will create a template for this project so it can be easily and quickly implemented across multiple other financial institutions in the future.

Maintenance: This stage will not apply to this project, as this application will not be in production at any institutions, however, it could be used in the future if this application were to be successfully implemented and bug fixes or enhancements were needed to satisfy a contract and any ongoing SLAs.

B4. Timeline and Milestones

Present a table showing for each milestone its projected start and end dates, and its projected duration:

Milestone	**Projected Start Date**	**Projected End Date**	**Duration (days/hours)**

Establish requirements for analytics process	12/27/2020	12/27/2020	1 day
Develop system workflow	12/28/2020	12/28/2020	1 day
Create Testing Dataset	12/29/2020	12/30/2020	2 days
Code application	12/31/2020	1/2/2021	3 days
Test application	1/3/2021	1/3/2021	1 day
Create user best practices guidelines	1/4/2021	1/4/2021	1 day

B5. Resources and Costs

Personnel, technology, or infrastructure	**Cost**
Card data including Card Number, Transaction Date, Merchant Name, Merchant ID, Terminal ID, and “Is-Fraud”.	N/A
Python development environment	N/A
72 Work hours	N/A

The resources needed for this project to be completed and implemented are limited to my time (72 hours’ worth of conception, design, coding, and testing), my computer for development (no cost, it has already been acquired), and a testing dataset (no cost other than time, as I am creating it myself). There are no additional resources or costs that will be associated with this project.

B6. Criteria for Success

The criteria for success I will use are three binary observations, which must all be returned correctly. In addition, a time comparison indicating statistical significance will also need to be satisfied. If all of these conditions are met, then this project will be deemed successful.

My application will need to correctly identify the compromised merchant from the test data, if the merchant is not detected, or a different merchant is detected, this is not a success.

The application will need to identify the correct date range of the merchant compromise. If the date range does not match, this is not a success.

The application will need to identify every card used at the compromised merchant during the compromise window. If all cards are not identified, this is not a success.

The application will need to complete the analytic process in a lower amount of time when compared to the average manual time for the same analysis. These results will be compared to determine if this difference in time is statistically significant. This project will be determined a success if the run time is proven to be lower by a statistically significant margin.

Criterion/Metric	**Required Data**	**Cut Score for Success**
Was proper merchant identified?	Output of Python application	Success is if and only if all data matches
Was proper date range identified?	Output of Python application	Success is if and only if all data matches
Were correct at-risk card numbers output?	Output of Python application	Success is if and only if all data matches
Time required to perform CPP	Amount of time for Python application to complete analysis, average manual analysis run time	Python application run time must be less than manual analysis run time by statistically significant margin

C. Design of Data Analytics Solution

C1. Hypothesis

Potential merchant compromises can be detected more quickly through automation than they could be detected by using traditional manual analysis.

C2. Analytical Method

The analytical method I will use in my data analysis solution will be descriptive. The data analysis technique I will use is data aggregation. I will aggregate my data based on different indicators in order to determine additional insights.

C2a. Justification of Analytical Method

I chose this method and technique because the objective of this application is to determine a trend across historical pieces of data, which is what the descriptive method provides. Data aggregation will allow for the different merchants and time frames to be evaluated to determine the merchant and date range where the cards were originally compromised.

C3. Tools and Environments of Solution

Python will be used to extract and manipulate the data because it will provide a robust environment to manipulate and analyze the dataset. Python will also provide an ideal means to output this compromise data in an easy-to-use format.

C4. Methods and Metrics to Evaluate Statistical Significance

The metric that will be used to determine statistical significance will be the run time of the Python application. This metric will be tested against the mean time it takes for the same analysis to be done manually. I will propose a null hypothesis, where the run time of the application is not significantly different than that of the manual average. I will then attempt to disprove that hypothesis, thereby proving that the application results in a statistically significant difference in analysis times. The values that will be used in this test to prove statistical significance are Z score, P-value, α =.01. If the P-value < α, I will be able to reject the null hypothesis and state that my application runs significantly faster.

C4a. Justification Of Methods and Metrics

This approach is the most appropriate because it will allow me to demonstrate that the difference in time is based on the process change that has occurred rather than sampling error.

C5. Practical Significance

The practical significance of the difference in run times will result in a significant amount of time being saved by the user. For example, if the application runs this analysis three minutes faster than the average time an analyst would take to complete the process manually, the tool will allow the user to reclaim three minutes of productivity each time they utilize the tool. If an analyst uses this tool three or more times per day, over a year, this could result in 1000 different instances where three minutes is saved, resulting in 3000 minutes, or 50 additional hours, the user can reallocate to other fraud mitigation activities.

C6. Visual Communication

I will choose a bell curve to represent my findings for statistical significance. A bell curve will most effectively demonstrate the distribution and standard deviation of the manual analysis. It will allow the plotting of the Z score, which will provide a good visual representation of the statistical significance of this metric. To visualize the practical significance, I will create a table that extrapolates the difference in time across different timeframes to illustrate the amount of time the application can save the user over a week, month, quarter, and year. A second table will be the best choice for my other binary variables to show whether or not the merchant name, compromise dates, and additional at-risk cards were properly output by the application.

D. Description of Datasets

D1. Source of Data

The source of my data is a pseudo-randomly generated dataset of card numbers, merchant names, transaction times, merchant ids, terminal ids, and an “is-fraud” datapoint. Given the extremely sensitive nature of credit card numbers, a dataset I create is the only way to ensure I do not inadvertently publish a legitimate card number or accuse a merchant of potential wrongdoing.

D2. Appropriateness of Dataset

The data is appropriate because it provides the proper fields for analysis within the application. Creating my dataset is the most appropriate way to provide my application with the data it will need while not relying on proprietary data, running afoul of payment card compliance regulations, or exposing myself to inadvertent legal ramifications through the publishing of a merchant breach without proper evidence.

D3. Data Collection Methods

To collect this data, I started by creating a set of 50 fictitious card numbers. I chose a number between 825 as the number of transactions per card with varying dates. I then created a fictitious list of 25 merchants, which were randomly assigned to the cards. I subsequently created a merchant id and terminal id for each of these merchants. I then chose three cards and inserted a fraudulent transaction to serve as the basis for the analysis, as these cards will be the initial inputs into my solution. I then manually performed the CPP analysis to ensure there was a common point of purchase across these three cards and to ensure there were additional cards in the CPP date range that would need to be pulled back by my application.

D4. Data Quality

Because I created the dataset, I was able to ensure that the data was exactly what would be needed for my application. This led to no formatting issues, dirty data, or other quality issues that would have needed to be addressed.

D5. Data Governance, Privacy and Security, Ethical, Legal, and Regulatory Compliance

A few factors led to the need to create my dataset. I did not feel comfortable using the card datasets I could find online due to ethical and privacy concerns. I would not want to contribute to a bad actor getting their hands on a legitimate payment card. In addition to these ethical and privacy concerns, PCI DSS compliance applies to payment cards, which prohibits card data from being sent in an insecure manner or being published where it could be accessed by unaffiliated third parties. Due to these concerns, I created my own card numbers and BINs (Bank Identification Numbers). A legal concern also impacted my dataset surrounding merchant names. I did not want to publish an analysis that accuses a real merchant of having a data breach, as this could be considered libel or slander given the medium.

D5a. Precautions

Out of precaution, I created a fictitious Bank Identification Number (999988) as well as 50 cards with different numbers for the last four digits of the card number. This will allow the application to distinguish the cards from one another while preventing any legitimate cards from being published. The other precaution taken was to create generic merchant names; by doing this, the conceptual identification of a compromised merchant could still be completed without accusing a specific merchant of suffering a data breach.

E. Sources

Krebs, B. (2014, March 5). Sally Beauty Hit By Credit Card Breach. Retrieved December 20, 2020, from https://krebsonsecurity.com/2014/03/sally-beauty-hit-by-credit-card-breach/

Holmes, I. (n.d.). How to uncover common point of purchase. Retrieved December 20, 2020, from https://www.sas.com/en_us/insights/articles/risk-fraud/common-point-of-purchase.html

Rippleshot Sonar. Retrieved December 20, 2020, from https://info.rippleshot.com/hubfs/Rippleshot_Sonar-Product_Sheet.pdf