Application of D-Cube: Detecting Fraudulent User Activity

17组 风险罗盘B组
顾胜达 孔文雁 刘田雨 李世鲲 潘俊廷 杨阁 杨梓溪 郑健

Introduction

As fraudsters continue to perfect the art of deception, fraudulent behaviors become less noticeable and more difficult to spot in large pools of user data. The question posed to us is, given a large multimodal data set of user activity and limited cases with known outcomes, how can we identify groups of fraudulent users and put a stop to their act?

Loosely synchronized activity, which occurs when fraudsters increase the number of fraud users and camouflage them hinders our ability to immediately identify fraudulent users and activity from regular ones. However, it is inevitable for fraud groups to display synchronized behavior on certain dimensions due to financial or resource constraints. The existence of dense blocks in user data often suggest coordinated fraudulent behavior, presented through common features such as similar timestamps, sharing resources, etc. Past studies have shown that dense blocks in real-world tensors (e.g., social media, Wikipedia, etc.) signal anomalous or fraudulent behavior such as retweet boosting, bot activities, and network attacks. The need to detect fraudulent behavior in large-scale multi-aspect data has driven the creation of D-cube, a disk-based dense block detection method which can also be run in a distributed manner across multiple machines. D-Cube identifies dense blocks in large-scale data that represent synchronized attacks from malicious sources. Other methods such as cluster analysis and boosting have also been used to identify synchronized fraudulent behaviors.

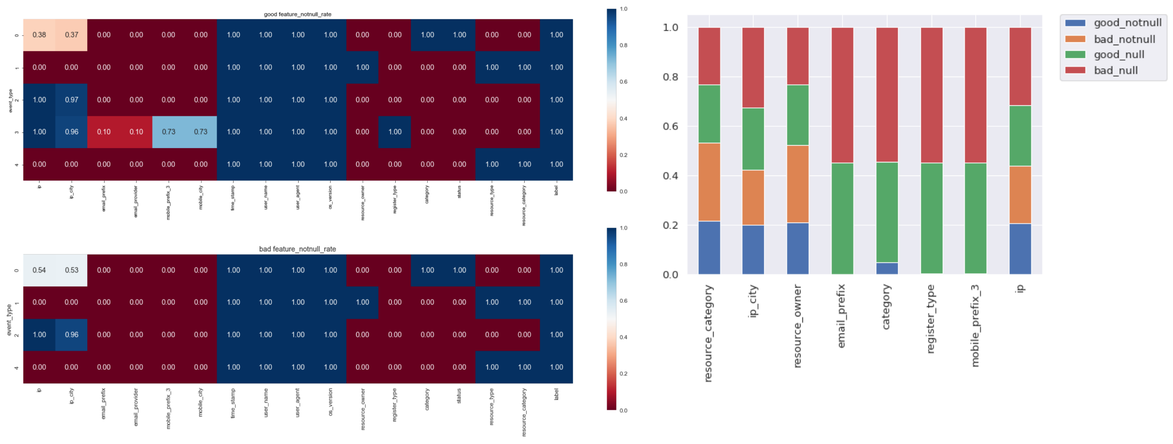
Objective

To analyze, cleanse, and apply the D-cube algorithm and other methods to a pool of multimodal data containing 13.8 million users and 53.8 million activity logs to detect groups of fraudulent activity and fraudulent users in our data.

Data Analysis

Missing data

Initial observations lead to our realization that large groups of missing data in particular features and events exist. The figures on the left below is a tally of the missing features sorted by event type and the nature of the user activity (e.g. Good/bad). The figure on the right shows the percentage of data existing for each event type out of the labeled events.



Good/Bad User Activity by Event Type and Features

The line graphs below visualize the number of times each event occurs along the time axis, where label 1 represents bad users and label 0 represent good users. The bar graphs represent the distribution of os_version and user_agent data.

Data Cleansing

Cleansing by Column

Local area networks (LAN) are set to null, ip_city dictionary encoded, ip address. Dictionary encoded, ip addresses and timestamps divided into sections.

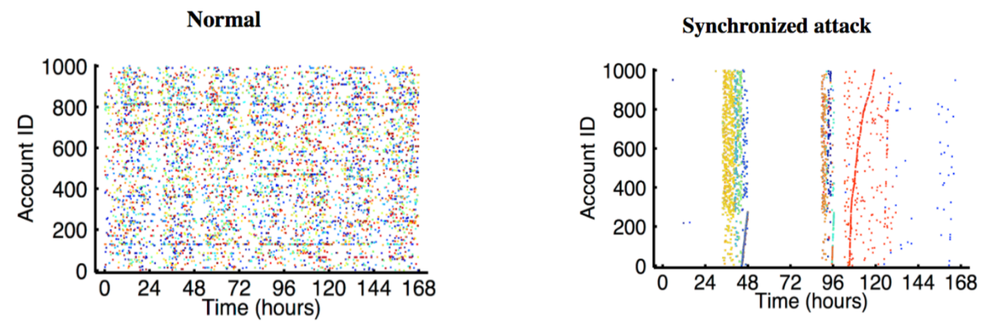
	time_stamp	time_stamp_day	time_stamp_hour	time_stamp_3hour	time_stamp_6hour
70310	2017-10-17 10:53:42	17	1710	1709	1706
70311	2017-10-18 22:02:50	18	1822	1821	1818
70312	2017-10-31 15:07:45	31	3115	3115	3112
70313	2017-10-30 19:03:53	30	3019	3018	3018
70314	2017-10-18 01:04:55	18	1801	1800	1800

Cleansing by Row

Deleting events 0, 1, 4 (not representative of the dataset), deleting repeated rows.

Key Traits

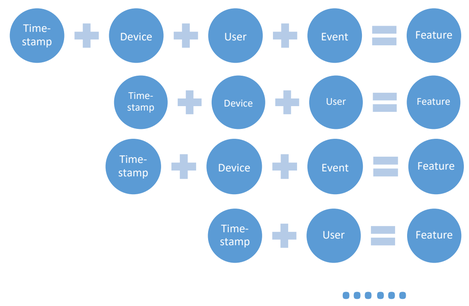
As seen in the figures below, normal user activity is scarce across multiple dimensions, whereas fraudulent attacks are synchronized in singular or multiple dimensions, causing denser regions of user data.



Feature Combinations

Feature_combination	Standard	Poly
[user_name, ip_123, time_stamp_6hour, event_type]	index	count
[user_name, ip_1234, time_stamp_day, event_type]	index	count
[user_name, ip_1234, time_stamp_hour, event_type]	index	count
[user_name, ip_1234, time_stamp_3hour, event_t...	index	count
[user_name, ip_1234, time_stamp_6hour, event_t...	index	count
[user_name, ip_city, time_stamp_day, event_type]	index	count
[user_name, ip_city, time_stamp_hour, event_type]	index	count
[user_name, ip_city, time_stamp_3hour, event_t...	index	count
[user_name, ip_city, time_stamp_6hour, event_t...	index	count
[user_name, resource_owner, time_stamp_day, ev...	index	count
[user_name, resource_owner, time_stamp_hour, e...	index	count

Suspicious user features (135 in total)



Feature_combination	Standard	Poly
[user_agent]	user_name	nunique
[user_agent, time_stamp_day]	user_name	nunique
[user_agent, time_stamp_hour]	user_name	nunique
[user_agent, time_stamp_3hour]	user_name	nunique
[user_agent, time_stamp_6hour]	user_name	nunique
[os_version]	user_name	nunique
[os_version, time_stamp_day]	user_name	nunique
[os_version, time_stamp_hour]	user_name	nunique
[os_version, time_stamp_3hour]	user_name	nunique
[os_version, time_stamp_6hour]	user_name	nunique
[ip_1]	user_name	nunique

Suspicious devices and events (175 in total)

From the above combinations, we selected 28 useful feature combinations to use in our model.

Method

Overall Structure of D-Cube

D-Cube is a search method that starts with the given relation and removes attribute values (and tuples with the attribute values) sequentially so that a dense block is left. Contrary to previous approaches, D-Cube removes multiple attribute values (and tuples with the attribute values) at a time to reduce the number of iterations and also the amount of disk I/O. In addition to this advantage, D-Cube carefully chooses attribute values to remove to give the same accuracy guarantee as if attribute values were removed one by one, and shows comparable or even higher accuracy empirically.

Single Block Detection

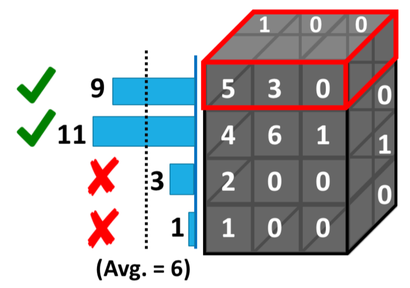
Step 1. search starts with the entire tensor.

Step 2. choose a mode with the most remaining slices.

Step 3. remove slices with mass at most the average mass.

Step 4. repeat until the tensor is empty.

*The detected dense blocks contain feature values that are suspected to be malicious. We fuse the dense blocks of 28 useful feature combinations to achieve the final results using XGBoost.



Results

Some of the results with D-Cube

TP 1849 P 5538 FP 903 N 3871
Test feature id: 42_3
Precision = 58.87%
Recall = 33.36%

TP 41 P 5484 FP 5 N 3828
Test feature id: 40_3
Precision = 85.1%
Recall = 0.74%

TP 937 P 5484 FP 304 N 3828
Test feature id: 64_1
Precision = 68.26%
Recall = 17.08%

TP 1106 P 5484 FP 311 N 3828
Test feature id: 149_2
Precision = 71.28%
Recall = 20.17%

The fusion results on a validation set with 1:1 good/bad samples

Th=0.7 Precision 71.0% Recall 45.8%
Th=0.6 Precision 65.9% Recall 70.4%
Th=0.5 Precision 62.4% Recall 87.0%
Th=0.4 Precision 60.2% Recall 90.0%

For more information, please scan the QR code below:

AlphaRisk

References

Shin, Kijung & Hooi, Bryan & Kim, Jisu & Faloutsos, Christos. (2017). D-Cube: Dense-Block Detection in Terabyte-Scale Tensors. 681-689. 10.1145/3018661.3018676.