|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Feature category | Original feature name | Original feature type | Transformation | No. of new features | New feature type |
| Transaction |  |  |  |  |  |
|  |  |  |  |  |  |
| Customer |  |  |  |  |  |
|  |  |  |  |  |  |
| Merchant |  |  |  |  |  |
|  |  |  |  |  |  |
| Location |  |  |  |  |  |

Assumption:

Every store can online and offline

'Store Credit Card', 'Cash Back Card', 'Premium Credit Card'

Specific merchant; Grocery Stores/Supermarkets, low price Jewelry Stores; Airline, high price Jewelry Stores

Card present

Compromised terminal🡪Credit Card Cloning – location information (customer x y, terminal x, y, merchant id)

Card not present

1, Fake QR code: A fake or unusual merchant name may appear in the transaction record for a specific period of time;

New stores are a prime target for fraud🡪 established date

2, quishing/account takeover/ etc:

odd customer IP address, odd transaction amt, product id vs card category

**Multiple Transactions**: The terminal might attempt to process multiple transactions in quick succession, sometimes with small amounts to test card validity ("test transactions").

IP address assume static, not dynamic for simplicity (requests to get location information?)

A specific period of time, merchant id 10 turns into fake id, with fake info

in python, for a credit card transaction dataframe, which was generated by a transaction class with attribute of fake merchant id, with below columns ['transaction\_date','customer\_id', 'amount','date\_in\_sec', 'No.\_of\_day', 'merchant\_id', 'pos\_id', 'merchant\_established\_date', 'MCCs', 'merchant\_category', 'IP\_address', 'type\_of\_credit\_card\_used', 'card\_present\_or\_not']

class imbalance (less than 1% of fraudulent transactions), a mix of numerical and categorical features (with categorical features involving a very large number of values), non-trivial relationships between features, and time-dependent fraud scenarios

Pick risk tolerance level

~~Transaction type: online/offline~~🡪IP address or null (null=offline)

Payment IP country

IP address country (cybercrime hubs)🡪count, one-hot

Blacked list IP🡪binary 1/0 (malicious IPs 111.111.111.111)

Product code e.g. limited edition product?

Email domain

New stores are a prime target for fraud🡪 established date

Short time multiple IP

~~Merchant id, merchant category~~ (fake QR code stuck on top of original QR)

Customer id, Card number (1 to many)🡪card number is PPI, changed it to credit card type

Same card ~~number~~ type, time since last transactions (in one single bank, 1 client max 1 card for each type)

Logic: Credit card type vs product id vs time since last transaction vs moving average amount?

Temporal features

Time anomaly🡪0-6, 6-0

Time seasonality🡪holiday

High freq transaction in short period of time

Device: phone, computer, other

Spending category transition vs transaction amount rapid change correlated?

Not used:

IP mismatch with shipping address (not good coz VPN; address potential PPI)

Concept drift

SHAP, mutual information

**1. Transaction Features**

These features describe the details of individual transactions:

* **Transaction Amount**: The dollar value of the transaction (normalized or bucketed).
* **Transaction Type**: (e.g., online, in-store, ATM withdrawal).
* **Merchant Category**: Type of merchant (e.g., electronics, groceries, luxury goods).
* **Currency**: Currency used in the transaction.
* **Time of Transaction**: Hour of day, day of week, or holidays (unusual times may indicate fraud).
* **Transaction Velocity**: Number of transactions within a specific time window (e.g., last hour/day).
* **Transaction Location**: Geographical location of the transaction.
* **Distance from User's Base Location**: Calculated distance between the transaction location and the user’s home address.

**2. User Behavior Features**

These features capture patterns in the user's historical behavior:

* **Average Transaction Amount**: Mean value of previous transactions by the user.
* **Transaction Frequency**: Number of transactions per day/week.
* **Preferred Merchant Categories**: Categories of merchants the user often shops from.
* **Historical Locations**: Regions or countries where the user commonly transacts.
* **Typical Device**: Devices the user frequently uses (e.g., smartphone, desktop).
* **Deviation from Normal Spending**: Difference between the current transaction and the user’s usual spending habits.

**3. Risk-Based Features**

These capture potential red flags:

* **IP Address Risk**: Risk score of the IP address used in the transaction (e.g., proxy/VPN use, flagged in blacklists).
* **Card Present vs. Card Not Present**: Was the card physically swiped, or was the transaction online?
* **Merchant Reputation**: Fraud history or risk score of the merchant.
* **Geographic Risk Score**: Risk level of the transaction's location based on known fraud hotspots.
* **Unusual Device/Browser**: Detecting if the transaction comes from a new or rare device.

**4. Temporal Features**

Patterns related to time:

* **Time Since Last Transaction**: Short time gaps between transactions may indicate fraud.
* **Transaction Time Anomaly**: Unusual transaction timing based on the user’s history (e.g., a transaction at 3 a.m.).
* **Burstiness**: High frequency of transactions in a short time window.

**5. Derived and Engineered Features**

Features created by combining or transforming raw data:

* **Spending Category Transition**: Abrupt changes in spending categories (e.g., groceries to luxury goods).
* **Cross-Border Transactions**: Transactions occurring in multiple countries within a short time frame.
* **Device Fingerprint**: Unique ID for the device used.
* **Spending Velocity**: Cumulative amount spent over a short period.

**6. External and Third-Party Data Features**

Augment internal data with external sources:

* **Geolocation Risk Data**: Fraud likelihood based on geolocation risk profiles.
* **Threat Intelligence**: Incorporate feeds on known hacker activity or blacklisted IPs.
* **Real-Time Fraud Scores**: Use third-party fraud detection APIs to generate additional risk scores.

**7. Fraud-Specific Indicators**

These are direct signals of fraudulent activity:

* **Chargeback Rate**: Number of chargebacks associated with the user/merchant.
* **Account Compromise Signals**: Indicators of compromised accounts, such as password resets or unusual login locations.
* **Mismatch in Shipping/Billing Address**: Addresses not matching may indicate fraud.

**8. Aggregate or Global Features**

Aggregated metrics across users or transactions:

* **Fraud Rate of Merchant**: Percentage of fraudulent transactions linked to the merchant.
* **Global Velocity**: Number of transactions made across all accounts/cards from the same IP address.

No KNN

Sequential processing

Challenging

Imbalance; panel, real time processing; explainability (financial regulation)

Dataset size, cost, time, batch/real time, explain for regulation,

0

We have two web applications that process payments using a payment processor. One web application is used by the customer service and the other one by a customer on the internet. When a payment needs to be processed the ip address of the client is sent in the payload to the payment processor. The payment processor has fraud detection rules enabled where if multiple payment calls are made from single ip then the ip is blocked. Now, when the customer service app makes the payment calls, they will send multiple calls for payment with the ip address of the system that the rep is using. This is causing an issue with the payment processor.