

Course Code: CSE 4460

**Course Name: Big Data Analytics
Lab**

Problem Statement

The main goal is to identify fraudulent transactions between bank customers by analyzing transaction data. We will accomplish this using only data visualization.

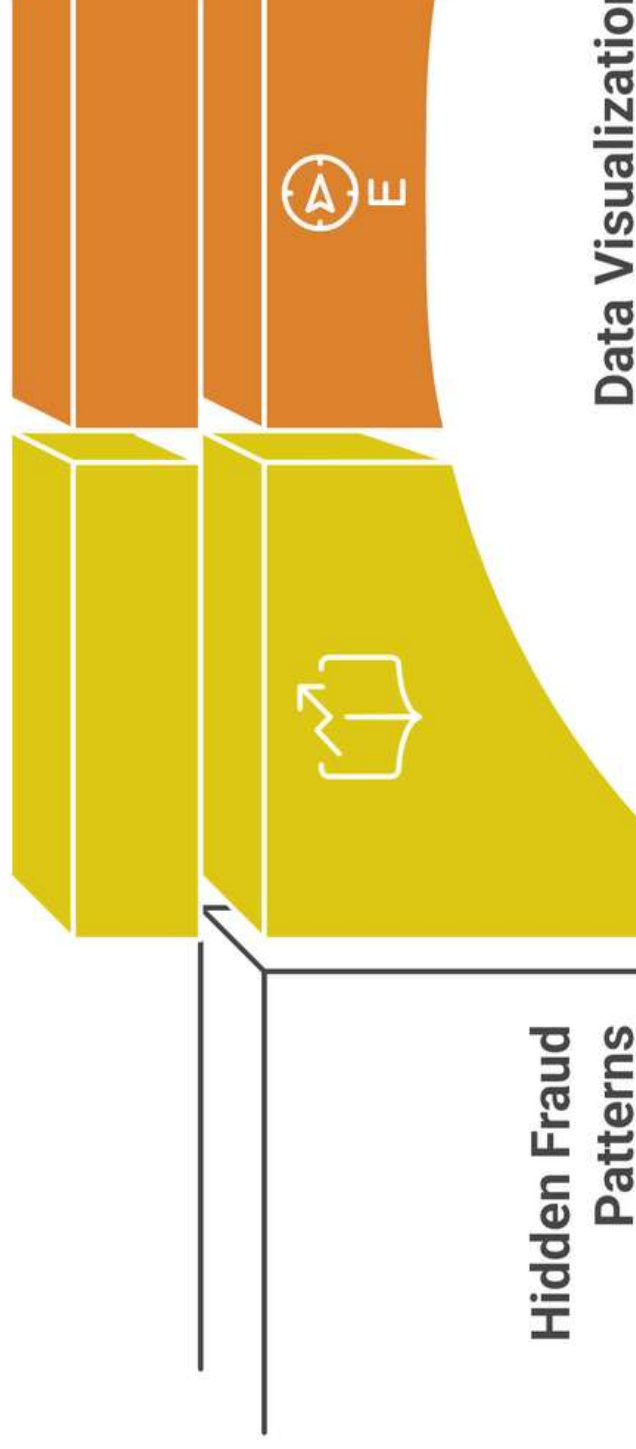
Using Visualization to See, Understand Transaction

Explainability

Visualize fraud for clear understanding

Exploration

Discover new fraud techniques



Data Visualization

Conceptual Idea to Detect

Identify Small Transactions



Transactions smaller
than average are
flagged

Analyze Transaction Patterns



The "Under the R

The core idea is that criminals often try to stay hidden by making their fraudulent transactions look insignificant. Large, unusual transactions (like buying a car in another country) are easy for banks' automated systems to flag as suspicious. To avoid this, criminals use a simple tactic: they make many small



The Camouflage

A "money mule" is a person who, knowingly or unknowingly, lets criminals use their legitimate bank account to move stolen money. The goal is to make the money harder to trace back to the original crime.

To look innocent, a mule account will often maintain a history of perfectly normal, everyday transactions. This creates a "behavioral baseline". The account might receive a regular salary, pay monthly bills, and have small, predictable expenses for groceries or coffee. It looks just like anyone else's account.

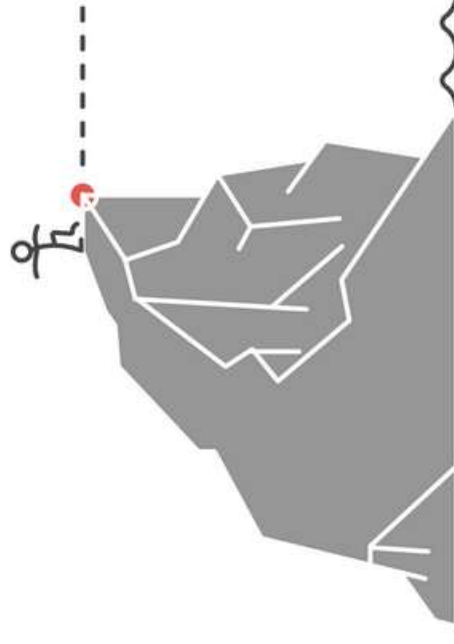
Normal Account Activity

Predictable, consistent financial behavior

Une

D

Large, appealing



The anomaly, or the red flag for fraud, is a

The "Cash-Out"

This rule focuses on a specific behavior: how criminals extract stolen money from the financial system and turn it into untraceable cash. After collecting money into a mule account, they need to withdraw it. To avoid suspicion, they don't make one large withdrawal. Instead, they make numerous small withdrawals or individual transfers over a short period.



Real Story

The amount of fraudulent transactions is smaller than the average amount of transaction by all users.

একটি ব্যাংক
ব্যক্তি তার
ছিল স্বাভাবিক
টাকার এক
এই গড় মাঝে
লেনদেন ক
সন্দেহজনক

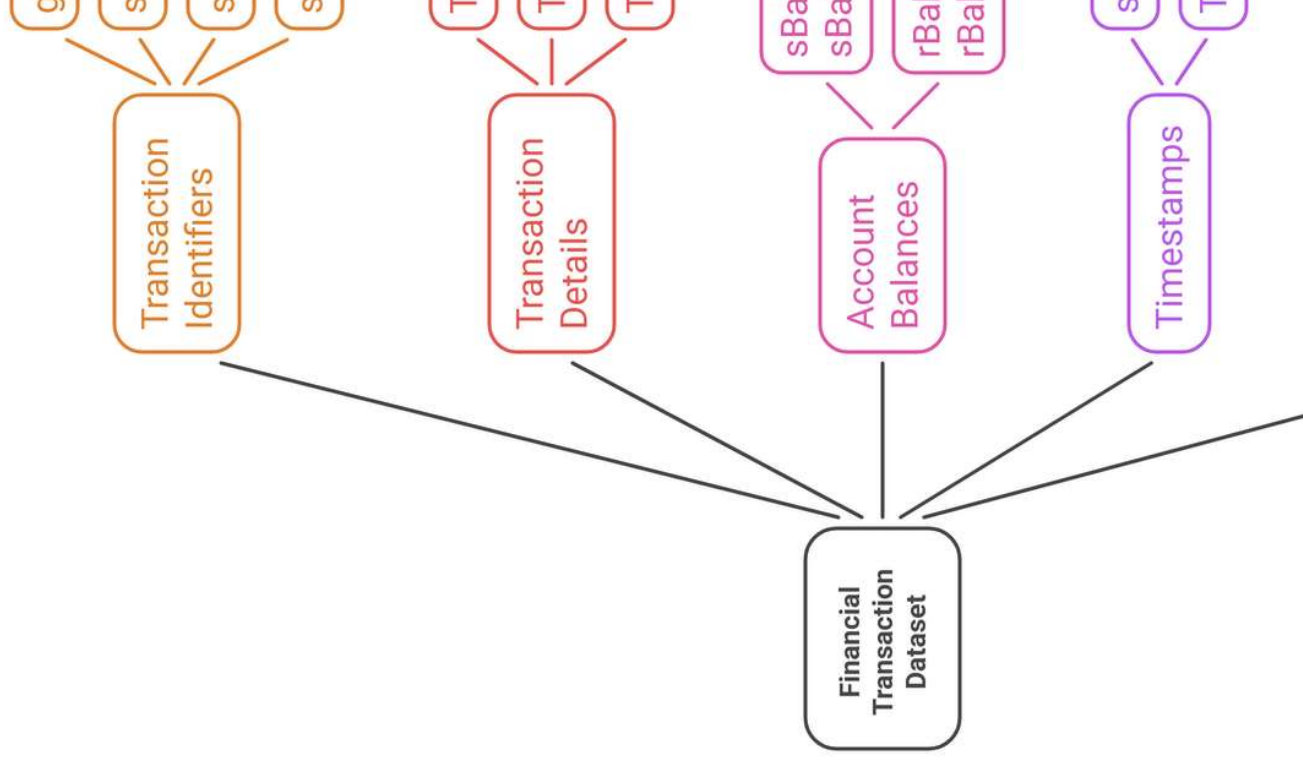
The mules can perform legitimate transactions but a sudden change in transferred money amounts corresponds to an anomaly.

'আসিফ'-এর
সপ্তাহে বাজ
ব্যালেন্স সব
একদিন, তার
এরপর, এই
নম্বরে ২০টি
আচরণের স
যাওয়া, এটা

An account holder that did several transactions for individual or withdrawal purpose with an amount lower than average amount of transaction can be considered as a fraudster and

একটি ব্যাংকে
যাক, যেখানে
করলে ব্যাংক
কৌশল অবলম্ব
ভিন্ন বিকাশ ব
পরিমাণ ৮,০০
'ক্যাস-আউট'

Dataset



Basic Characteristics

This is a critical characteristic for data cleaning and imputation:

- **54222 non-null** : The majority of your columns are **complete**, meaning they have no missing values for any of the 54,222 records. This is great for analysis as you won't need impute or drop rows based on these columns. These columns include: `gT`, `sID`, `rID`, `sAcc`, `rAcc`, `TranAmount`, `TranType`, `TranStatus`, `sBalBefore`, `sBalAfter`, `rBalBefore`, `rBalAfter`, `sf1`, `sf2`, `sTD`, `rTD`, `sAccID`, `NoDescription`, `TranTS`, `sType`, `rType`.
- **0 non-null** : Four columns, specifically `sf3`, `sf4`, `ef1`, and `ef2`, have **0 non-null** entries. This means **all 54,222 entries in these four columns are null or empty**. This is a very strong indicator that these columns are entirely useless for your analysis and should be dropped, as you noted in your previous context.

It is observed that service fields 3,4, and empty fields 1 and 2 have no values.

Data Preparation

```
#So drop the empty columns and columns with duplicate data.  
columns = ['sf3', 'sf4', 'ef1', 'ef2', 'sAccID', 'NoDescription']  
df.drop(columns, inplace=True, axis=1)  
data = df
```

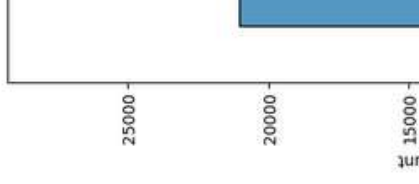
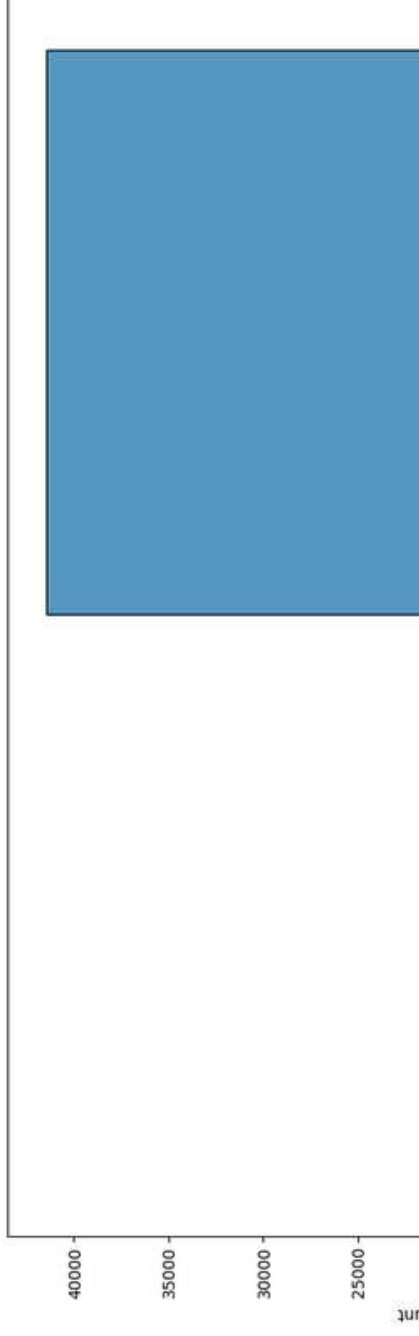
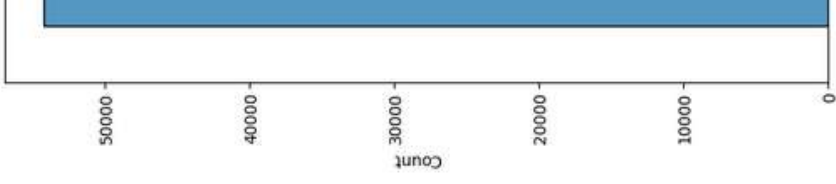
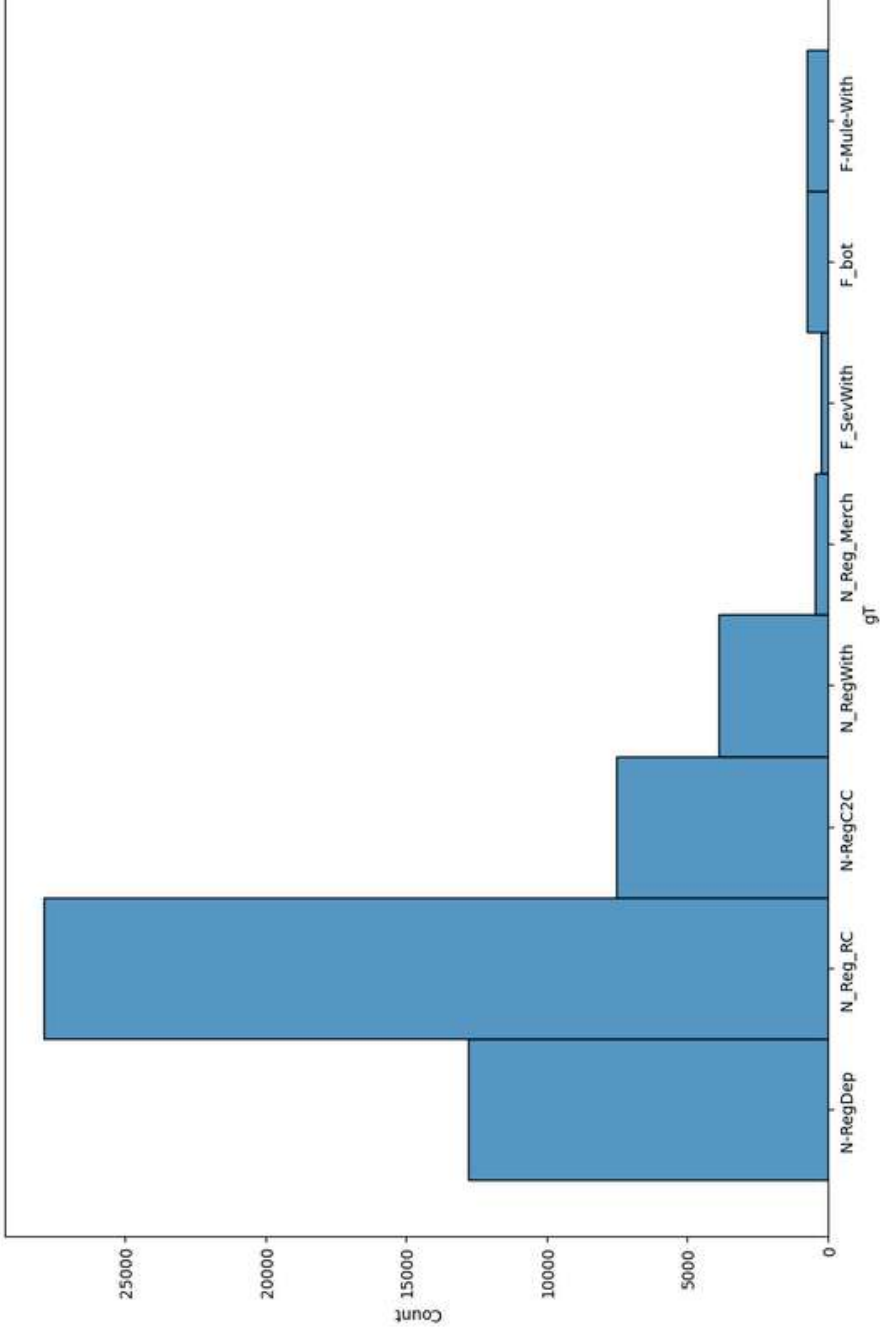
Data Analysis

0

gT	N-RegDep	N_Reg_Dep
sld	PN_Ret2	PN_EU_1_50
rld	PN_EU_0_261	operator
sAcc	RAcc2	EUAcc1_50
rAcc	EUAcc0_261	Acc
TranAmount	131926.49	2054.49
TranType	Dt	ArRt
TranStatus	SU	SU
sBalbefore	1000000000.0	100000000.0
sBalAfter	999868073.51	99997945.51
rBalBefore	100131926.49	99180036.51
rBalAfter	1000000000.0	99177982.51
sf1	True	True
sf2	True	True
sTD	1/6/2011 0:11:22	1/6/2011 0:16:30
rTD	1/6/2011 0:11:22	1/6/2011 0:16:30

```
df2=np.transpose(data)  
df2
```

Data Analysis



Data Analysis

1. gT (Grouped Transaction Type)

mathematica

N_Reg_RC	27,981
N-RegDep	12,784
N-RegC2C	7,584
N_RegWith	3,899
F_bot	731
F-Mule-With	729
N_Reg_Merch	442
F_SevWith	232



Insight:

- The majority of transactions are normal (N_Reg_RC, N-RegDep, N-RegC2C, N_RegWith, N_Reg_Merch, F_bot, F-Mule-With, F_SevWith) are normal (54,000+).
- Fraudulent transactions (F_bot, F-Mule-With, F_SevWith) are normal (54,000+).

This shows a clear distinction between normal transactions, which are common in the dataset, and fraudulent transactions, which are rare.

Data Analysis

2. TranStatus

nginx	
SU	54,222



Insight:

- Every transaction is marked as successful (SU).
- The dataset does not contain failed transactions (filtered out).

Data Analysis

3. sType (Sender Type)

nginx

EU	41,438
RET	12,784



Insight:

- Most senders are End Users (EU).
- A smaller portion are Retail accounts (RET).

Data Analysis

4. rType (Receiver Type)

```
cpp
operator 27,901
EU       21,019
RET      4,860
MER       442
```

✓ Insight:

- Most receivers are operators or other end users.
- Very few are retailers or merchants (MER).
- This suggests the system mainly facilitates person-to-person and merchant-based payments.

Data Analysis, Continued

- Dataset size: 54,222 successful transactions.
- Fraudulent cases: ~1,700 (3%) → very imbalanced data.
- Senders: mostly end users.
- Receivers: mostly operators and end users.
- Merchant transactions are rare, which may be unusual.

Building the C

Part	What the Code Does	Purpose
1	Convert dataframe columns (<code>sAcc</code> , <code>rAcc</code> , <code>TransAmount</code>) into Python lists. Count total and unique senders/receivers.	Prepare data for graph creation and determine node size & type.
2	Create a directed transaction graph (<code>DiGraph</code>) with 1000 edges. Each edge = one transaction (sender → receiver). Edge thickness = transaction amount.	Visualize transaction flow.
3	Create an undirected graph (<code>Graph</code>) with all transactions. Edge weights = transaction amounts. Add hover tooltips.	Broader view of transaction network.
4	Build another graph (<code>G2</code>) with all transactions again.	Main visualization.
5	Calculate node degrees (number of connections). Select accounts with ≥10 transactions as "repeated nodes."	Filters out low-frequency accounts to highlight fraud suspects.

Building the C

7

Create a new dataframe `df2` containing only transactions from possible fraud accounts.

NaN

8

From `df2`, drop transactions that are not of type `Individual` or `Withdrawal`. Keep only these two types.

Fraud detection

9

Calculate the average transaction amount. Mark transactions with amount \geq average for removal.

Fraud

10

Drop transactions above average → keep only smaller-than-average transactions in `df4`.

NaN

11

Print number of transactions left in `df4`.

Show

12

Visualize graph again with PyVis:

Final

- Normal transactions (first 1000) in gray.

action

How the Graph Connects

Concept	Meaning	Where in Code
1. Fraudulent transactions are smaller than the average transaction amount	Fraud often hides in <i>small below-average transfers</i> .	Part 9 & Part 10
2. Mules can perform legitimate transactions but sudden change in amounts = anomaly	Mule accounts look normal, but unusual patterns (e.g., sudden shifts, many transactions) are suspicious.	Part 5–7
3. Fraudsters = accounts doing multiple Individual/Withdrawal transactions below average amount	Accounts repeatedly doing small “Ind” or “WI” transactions are likely fraud.	Part 8 & Part 9

How the Graph Connects

Concept	Meaning	Where in Code
1. Fraudulent transactions are smaller than the average transaction amount	Fraud often hides in <i>small below-average transfers</i> .	Part 9 & Part 10
2. Mules can perform legitimate transactions but sudden change in amounts = anomaly	Mule accounts look normal, but unusual patterns (e.g., sudden shifts, many transactions) are suspicious.	Part 5–7
3. Fraudsters = accounts doing multiple Individual/Withdrawal transactions below average amount	Accounts repeatedly doing small “Ind” or “WI” transactions are likely fraud.	Part 8 & Part 9

Why No ML/DL Was

1. Exploratory Phase

- The notebook's goal was to explore data and understand fraud
- Instead of training a classifier, you applied domain-driven rules (transaction amount).

2. Severe Class Imbalance

- Fraudulent cases are < 3% of total transactions.
- Training ML/DL directly on such skewed data without handling (e.g., oversampling or synthetic data generation) would lead to a biased model that just predicts "no fraud."

3. Graph Structure

- Transactions were modeled as a network (graph).
- ML/DL would need Graph Neural Networks (GNNs) or embedding techniques (e.g., Node2Vec, DeepWalk) to handle graph structure. That's more complex than traditional ML.

4. Interpretability

- The rule-based method (small amount, frequent transactions)

Referenc

- [1] Evgenia Novikova, Igor Kotenko and Evgenii Fedotov. Interactions of Mobile Money Transfer Services, International Journal of Mobile Computing, 6(4), 73-97, October-December 2011.
- [2] Rieke, R., Zhdanova, M., Repp, J., Giot, R., & Gaber, C. (2013). The 2nd International Workshop on Human-Computer Interaction for Process Behavior Analysis. The 2nd International Workshop on Human-Computer Interaction for Process Behavior Analysis. The 2nd International Workshop on Human-Computer Interaction for Process Behavior Analysis. Management (RaSIEM 2013) (pp. 66-77). Springer.

THE