Scope
Volume 14 Number 01 March 2024

Synthetic Financial Datasets for Fraud Detection: Exploring Robust Models and Techniques to Tackle Cloud and Mobile Computing Challenges

Pankaj Agarwal
Professor & Dean, School of Engineering & Technology
K.R Mangalam University

1. **Abstract:**

The key worry for several sectors, including the government and consumers is financial fraud. Cloud computing and mobile computing have created more issues recently. Conventional manual detection methods take a lot of time, are inaccurate, and can't manage massive data on their own. Hence, a variety of methods have been used to address this extremely important issue of financial fraud. Instead of being produced by actual events, "Synthetic Financial Datasets for Fraud Detection" is synthetic data that has been created. Due to the confidentiality of financial services information, it was developed utilizing the mobile money payment simulator (PaySim). Customer and fraudulent behavior are present in the data produced by the simulator. The management of this data would be difficult because of its larger magnitude. This work has addressed different types of financial frauds involved during the transactions. The exploratory data analysis is applied to explore the features. Dataset is quite huge & unbalanced to process on conventional machines and therefore various sampling techniques were explored to balance the dataset for the better results in terms of accuracy and make the data set reliable. Dataset is divided into 15 chunks with 12 chunks for training and 3 chunks for testing purpose. Various classification techniques including ensemble techniques, Ada Boost, decision tree have been applied on each of the chunk. To ensure the reliability of the model, the results were compared with ensemble technique and decision tree classifier. With feature selection & dataset balancing, the model is showing 80 percent of accuracy.

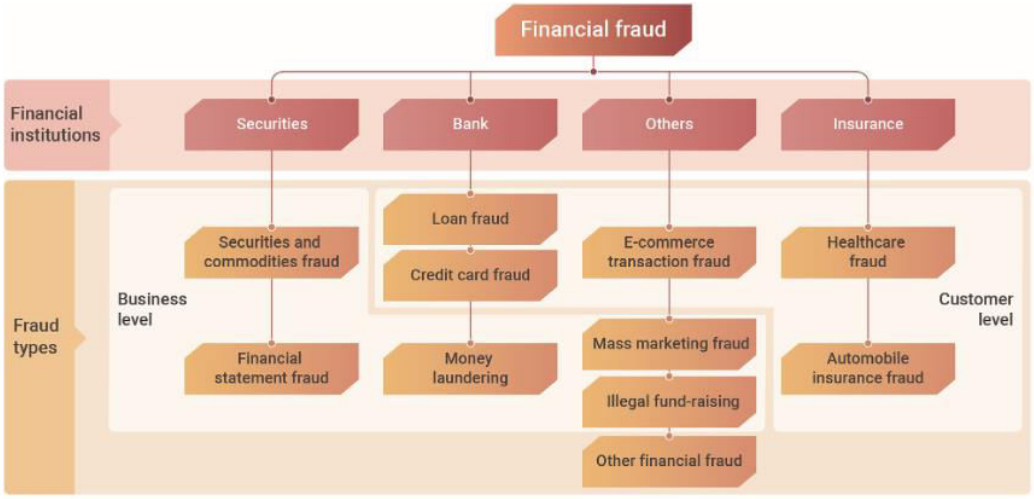
Keywords: Financial fraud, big data, fraud detection, synthetic data, simulator, exploratory data analysis (EDA), sampling, ensemble technique, ada boost, decision tree, classification, feature selection.

2. Introduction

Financial fraud is the main concern for many industries from government to consumers. In recent times, cloud computing and mobile computing has added more problems. Traditional methods involving manual detection is tome consuming, not accurate and also impossible to handle big-data manually. So various techniques have been in place to resolve this very crucial problem in financial fraud.

"Intelligent financial fraud detection practices in post-pandemic era" paper has addressed the types of financial frauds, types of data used for fraud detection and current practices in different industries.

Financial fraud is classified into different types securities, bank, e-commerce transaction fraud, insurance including Healthcare and Automotive insurance fraud and others. Form the figure shown below we can identify the types of financial frauds on different industries.



3. Literature Survey:

In one of the publications (Applying simulations to the problem of detecting financial fraud) Edgard Alonso Lopez-Rojas introduced a financial simulation model encompasses of two financial domains: Mobile payments and retail store systems. The objective of this project is to use computer-based simulation for fraud detection in financial domains. Author used two different simulators to generate a synthetic dataset which includes normal and fraudulent transactions and they are: Payment Simulator (PaySim) and Retail store (RetSim) simulator.

Using the statistical approach and social network analysis (SNA) on real data author compared the relations between agents and generated synthetic data. Paper shows the

effective way of identifying fraud also helped managers to priorities the fraud detection and how to invest in fraud detection

In this particular project the payment simulator dataset is taken from Kaggle and referred the work of E. A. Lopez-Rojas and S. Axelsson[1]. There are different types of algorithms used in financial fraud detection, in this project with the use of classification models like Logistic regression, KNN, Decision tree classification, naive bayes, NN, ensemble techniques and Random Forest. Generic process flow has been used from the Supervised Machine Learning Algorithms [2]

Logistic regression:

This classification method employs a single multinomial logistic regression model with a single estimator and builds its model using the class. In a particular way, logistic regression often identifies the location of the boundary between the classes and notes that class probabilities vary with proximity to the boundary. With a larger data set, this advances more quickly toward the extremes (0 and 1). These probabilistic claims are what differentiate logistic regression from simple classifiers. It can be fitted differently and provides predictions that are stronger and more specific, but those precise forecasts could be off. Like Ordinary Least Squares (OLS) regression, logistic regression is a method of prediction. But with logistic regression, the outcome of the prediction is (dichotomous Newsom, I. (2015).

Naive Bayes Networks:

These are very basic Bayesian networks, consisting of directed acyclic graphs with just one parent (representing the unobserved node) and a number of children (corresponding to observed nodes), with a strong assumption of independence among child nodes in the context of their parent [3]

Decision tree:

Instances are classified using Decision Trees (DT), which sort instances according to feature values. In a decision tree, each node represents a feature in an instance that needs to be classified, and each branch represents a possible value for the node. Beginning at the root node, instances are categorised and arranged according to the values of their features [4]. Observations about an item are mapped to conclusions about the item's target value using a decision tree as a predictive model in decision tree learning, which is used in data mining and machine learning.

Neural Networks:

The input and activation functions of the unit, the network design, and the weight of each input link are the three main components that determine the performance of an artificial neural network (ANN). The function of the ANN is determined by the weights' current values because the first two aspects are fixed.

Although typically each network only performs one, [5] opined Neural Networks (NN) that can actually conduct a number of regression and/or classification tasks at once. Consequently, the network will often only have one output variable, although this may correspond to multiple output units in the case of many-state classification issues (the post-processing stage takes care of the mapping from output units to output variables).

K means:

K means is considered one of the simplest unsupervised learning algorithms that addresses the well-known clustering problem [6,7]. The process uses a predetermined number of clusters (let's assume k clusters) fixed a priori to categories a given data set. When labelled data is not available, the K-Means technique is used

Ensemble Technique

Multiple models are used in ensemble approaches to improve performance. Several research areas, including computational intelligence, statistics, and machine learning, have used ensemble approaches. The ensemble methods are divided into traditional ensemble methods like bagging, boosting, and random forest, as well as deep learning-based ensemble methods, multi-objective optimization-based ensemble methods, fuzzy ensemble methods, and methods based on multiple kernel learning and negative correlation learning [8].

4. Review of Data Set

There is a dearth of publicly available datasets on financial services and especially in the emerging domain of mobile money transactions.

In this work the payment simulator dataset is taken from Kaggle and referred the work by E. A. Lopez-Rojas and S. Axelsson. “Multi Agent Based Simulation (MABS) of Financial Transactions for Anti Money Laundering (AML)”. We have taken the reference of a synthetic dataset generated using the PaySim simulator available on kaggle. PaySim uses

aggregated data from a private dataset to create a synthetic dataset that resembles normal transaction traffic and injects malicious behavior to later evaluate the performance of fraud detection methods (LOPEZ-ROJAS, 2016)

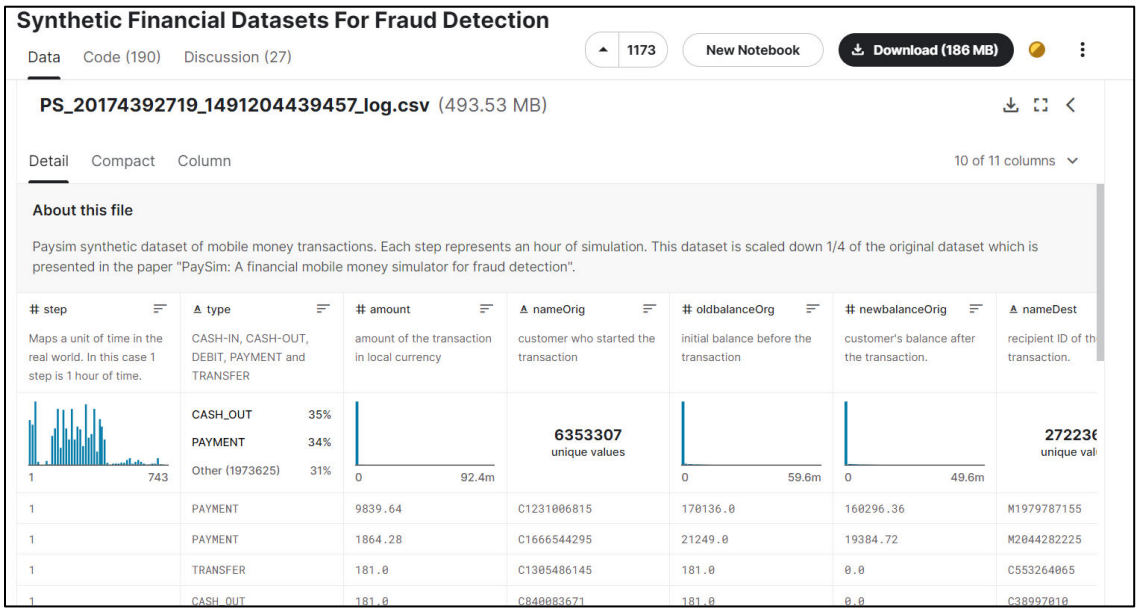


Fig 1: Reference to Kaggle data Source

The dataset consists of 11 variables and total number of observations is 63,62,620. The variables and their descriptions are as follows:

- step column: Number of hours it took for a transaction to complete.
- type column: Type of transaction that took place. There are 5 categories in this column namely; 'PAYMENT', 'TRANSFER', 'CASH_OUT', 'DEBIT', 'CASH_IN'.
- nameOrig: Name/ID of the Sender.
- oldbalanceOrg: Sender balance before the transaction took place.
- newbalanceOrg: Sender balance after the transaction took place.
- nameDest: Name/ID of the Recipient.
- oldbalanceDest: Recipient balance before the transaction took place.
- newbalanceDest: Recipient balance after the transaction took place.
- isFraud: This is the transactions made by the fraudulent agents inside the simulation. **(Target Variable)**

The business model aims to control massive transfers from one account to another and flags illegal attempts. An illegal attempt in this dataset is an attempt to transfer more than 200.000 in a single transaction.

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
0	1	PAYMENT	9839.64	C1231006815	170136.0	160296.36	M1979787155	0.0	0.0	0	0
1	1	PAYMENT	1864.28	C1666544295	21249.0	19384.72	M2044282225	0.0	0.0	0	0
2	1	TRANSFER	181.00	C1305486145	181.0	0.00	C553264065	0.0	0.0	1	0
3	1	CASH_OUT	181.00	C840083671	181.0	0.00	C38997010	21182.0	0.0	1	0
4	1	PAYMENT	11668.14	C2048537720	41554.0	29885.86	M1230701703	0.0	0.0	0	0

Fig 2: Top 5 financial dataset

```
data.shape
```

```
(6362620, 11)
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6362620 entries, 0 to 6362619
Data columns (total 11 columns):
 #   Column          Dtype
---  ------          -----
 0   step            int64
 1   type            object
 2   amount          float64
 3   nameOrig        object
 4   oldbalanceOrg   float64
 5   newbalanceOrig  float64
 6   nameDest        object
 7   oldbalanceDest  float64
 8   newbalanceDest  float64
 9   isFraud         int64
 10  isFlaggedFraud  int64
dtypes: float64(5), int64(3), object(3)
memory usage: 534.0+ MB
```

Fig 3: Data shape and Datatype

```
data.describe()
```

	step	amount	oldbalanceOrg	newbalanceOrig	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
count	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06
mean	2.433972e+02	1.798619e+05	8.338831e+05	8.551137e+05	1.100702e+06	1.224996e+06	1.290820e-03	2.514687e-06
std	1.423320e+02	6.038582e+05	2.888243e+06	2.924049e+06	3.399180e+06	3.674129e+06	3.590480e-02	1.585775e-03
min	1.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
25%	1.560000e+02	1.338957e+04	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
50%	2.390000e+02	7.487194e+04	1.420800e+04	0.000000e+00	1.327057e+05	2.146614e+05	0.000000e+00	0.000000e+00
75%	3.350000e+02	2.087215e+05	1.073152e+05	1.442584e+05	9.430367e+05	1.111909e+06	0.000000e+00	0.000000e+00
max	7.430000e+02	9.244552e+07	5.958504e+07	4.958504e+07	3.560159e+08	3.561793e+08	1.000000e+00	1.000000e+00

```
data.isnull().sum()
```

```
step              0
type              0
amount            0
nameOrig          0
oldbalanceOrg     0
newbalanceOrig    0
nameDest          0
oldbalanceDest    0
newbalanceDest    0
isFraud           0
isFlaggedFraud    0
dtype: int64
```

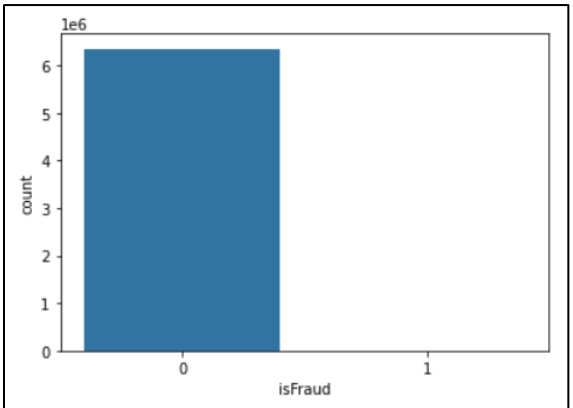
Fig 4: Data summary and null values

From the above figure we can see that the shape of the dataset is 63,62,620X11, there are mix of data types in the dataset. Further down the line during data-preprocessing these kinds of datatypes has been converted or eliminated during data type conversion/feature

scaling and selection. isFraud is considered as the target variable. Also, to understand the data before pre-processing, missing values has also been checked.

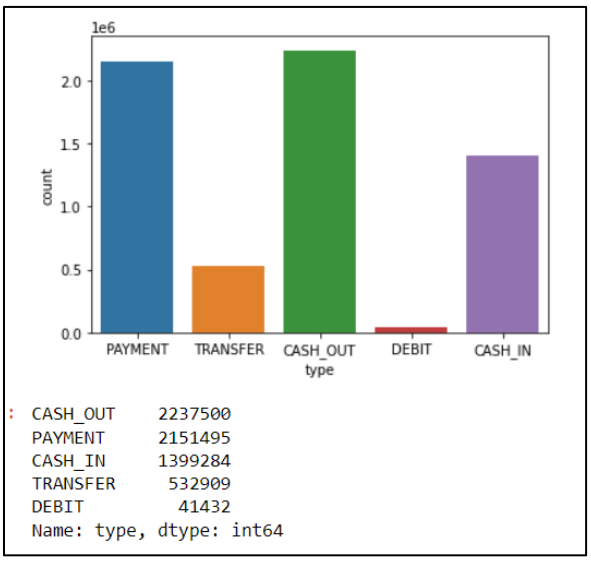
5. Data Pre-Processing:

Data pre-processing has been carried out on the data set to evaluate different key parameters. Is Fraud being the target variable and initial visualization has been performed to understand the pattern and behavior of the dataset. From the below bar plot, we can see the number of frauds and non-frauds available in the dataset, overall there were 63,54,407 non-fraud and 8213 fraud data were found. As we can see there were huge variations in other words, we can say there are imbalance in the data set. Accordingly, sampling technique has been applied on the dataset to have a proper and reliable accuracy during modeling stage.




```
0    6354407
1       8213
Name: isFraud, dtype: int64
```

As mentioned above there were four different types of transaction in the dataset. Based on the plot the payment and cash-out has the higher transaction when compared to transfer and debit. Also, we can see there are some decent numbers of transaction happed in the cash-in type. Its is evident that more than 50% of transaction happened in cash out, payment and cash in.



As a next step of the process, we need to check the missing values and treat outliers if there are any. It is found that there are no missing values. So, it is not required to perform the missing value analysis for this dataset. From the below figure we can see there are number of outliers present in the dataset and the variables that posses' outliers are "amount", "old balance org", "new balance org", "old balance dest", "new balance dest". These datapoints involved with currency transaction, it would be unappropriated to perform the outlier analysis on the dataset as these transactions are based on individual interest.

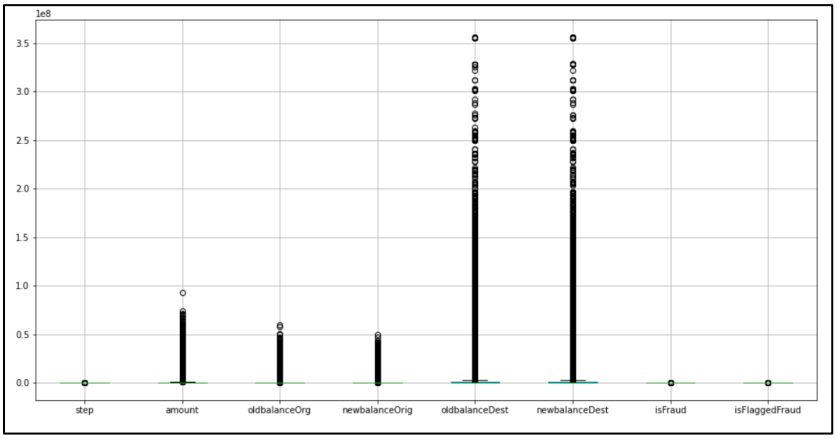


Fig 5: Outliers (Box Plot)

Feature selection has been carried out to see the relationships and correlation between each continuous variable. Form the below table we can see the positive and negative correlation between the feature. Moreover, there is strong positive correlation between "old balance org", "new balance org", "old balance dest", "new balance dest".

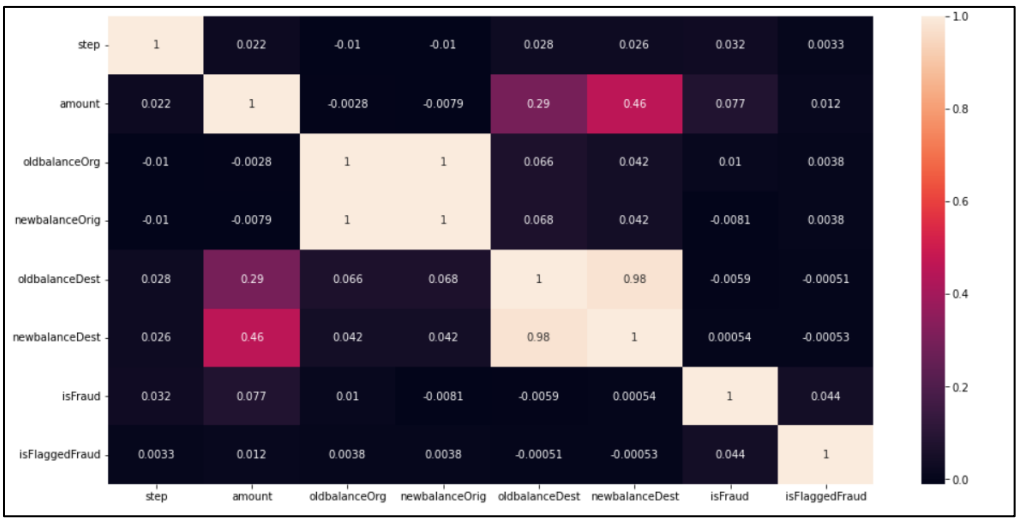


Fig 6: Correlation matrix (Heat Map)

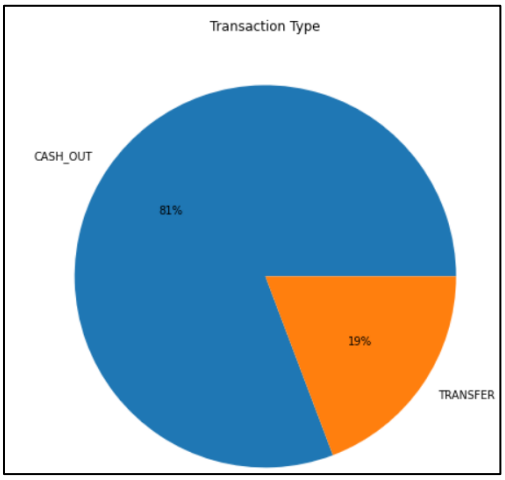
Moving further, “old balance org” and “new balance dest” has been dropped from the dataset to reduce the complexity of the project. The data has been grouped with the transaction type on basis of count.

type	step	amount	nameOrig	newbalanceOrig	nameDest	oldbalanceDest	isFraud	isFlaggedFraud
CASH_IN	1399284	1399284	1399284	1399284	1399284	1399284	1399284	1399284
CASH_OUT	2237500	2237500	2237500	2237500	2237500	2237500	2237500	2237500
DEBIT	41432	41432	41432	41432	41432	41432	41432	41432
PAYMENT	2151495	2151495	2151495	2151495	2151495	2151495	2151495	2151495
TRANSFER	532909	532909	532909	532909	532909	532909	532909	532909

Overall, non-fraud and fraud has already been discussed in the above paragraph, analysis has been done to check the number of frauds at each transaction type and from the below figure we can see that the fraud has identified at only two transaction type and they are **cash-out** and during **transfer**

isFraud type	0	1
CASH_IN	1399284.0	NaN
CASH_OUT	2233384.0	4116.0
DEBIT	41432.0	NaN
PAYMENT	2151495.0	NaN
TRANSFER	528812.0	4097.0

Altogether there are 81% of fraud has occurred in cash-out type and 19% occurred during transfer.



Dataset has been further filtered out only two transaction types where the fraud is identified and the label encoding has been done for those variable as we can see in the below figures.

isFraud	0	1
type		
CASH_OUT	2233384	4116
TRANSFER	528812	4097

isFraud	0	1
type_encoded		
0	2233384	4116
1	528812	4097

	step	type	amount	nameOrig	newbalanceOrig	nameDest	oldbalanceDest	isFraud	isFlaggedFraud	type_encoded
2	1	TRANSFER	181.00	C1305486145	0.0	C553264065	0.0	1	0	1
3	1	CASH_OUT	181.00	C840083671	0.0	C38997010	21182.0	1	0	0
15	1	CASH_OUT	229133.94	C905080434	0.0	C476402209	5083.0	0	0	0
19	1	TRANSFER	215310.30	C1670993182	0.0	C1100439041	22425.0	0	0	1
24	1	TRANSFER	311685.89	C1984094095	0.0	C932583850	6267.0	0	0	1

Additionally, features like “isFlaggedFraud”, “nameOrig”, “nameDest”, “step”, “type” (as encoded) as these variables does not possess any values or the information that helps in modelling. By dropping these variables, the complexity/dimension of the data is further reduced which will eventually helps in processing speed and better accuracy in terms of prediction. Now the shape of the dataset has been dropped from **6362620,11 to 2770409, 5**. The below figure shows the total variables after feature selection.

	amount	newbalanceOrig	oldbalanceDest	isFraud	type_encoded
2	181.00	0.0	0.0	1	1
3	181.00	0.0	21182.0	1	0
15	229133.94	0.0	5083.0	0	0
19	215310.30	0.0	22425.0	0	1
24	311685.89	0.0	6267.0	0	1

Fig 7: Final Variables

After dropping all the unnecessary variables, the total number of fraud and non-fraud has been checked as shown below. This helped us to proceed further to perform sampling as the data is heavily imbalanced.

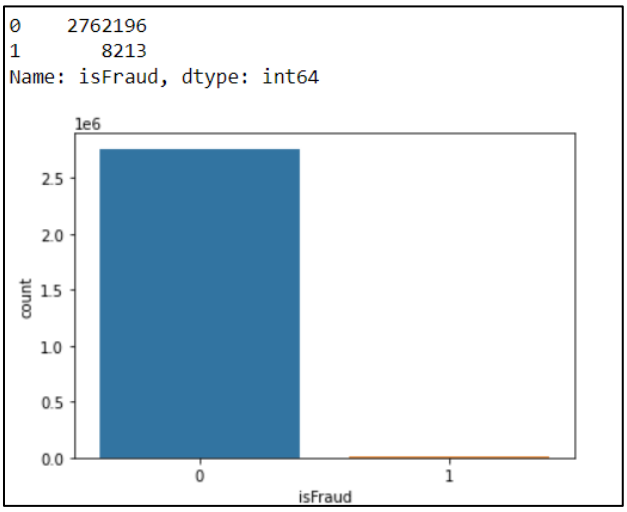
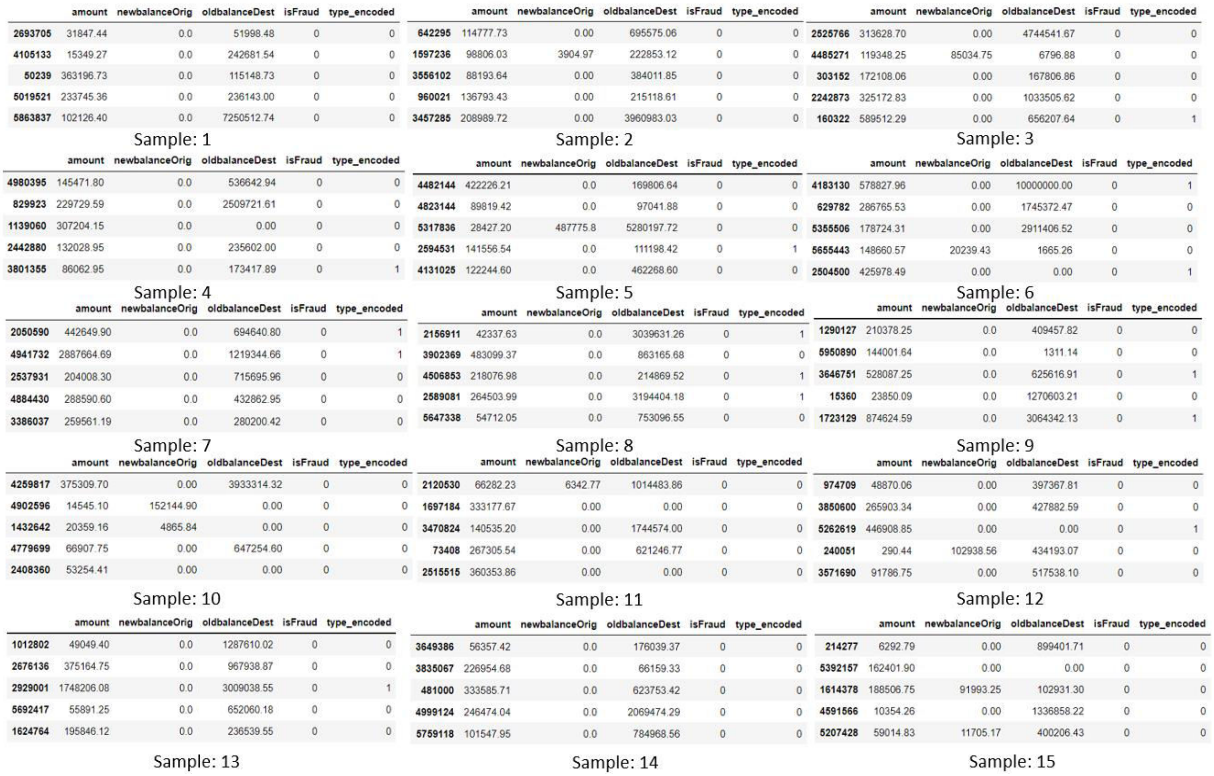


Fig 8: Fraud after dimensionality reduction

6. Sampling

To balance the dataset, simple random sampling has been applied where 50% of the data has been taken from fraud and 50% with non-fraud. This will help algorithms to predict the fraud in future with high accuracy. Total observation has been divided into 15 different samples with 16,000 observations in each as shown below.



Sample: 1

	amount	newbalanceOrig	oldbalanceDest	isFraud	type_encoded
2693705	31847.44	0.0	51998.48	0	0
4105133	15349.27	0.0	242681.54	0	0
50239	363196.73	0.0	115148.73	0	0
5019521	233745.36	0.0	236143.00	0	0
5863837	102126.40	0.0	7250512.74	0	0

Sample: 2

	amount	newbalanceOrig	oldbalanceDest	isFraud	type_encoded
642295	114777.73	0.00	695575.06	0	0
1597236	98806.03	3904.97	222853.12	0	0
3556102	88193.64	0.00	384011.85	0	0
960021	136793.43	0.00	215118.61	0	0
3457285	208989.72	0.00	3960983.03	0	0

Sample: 3

	amount	newbalanceOrig	oldbalanceDest	isFraud	type_encoded
2525766	313628.70	0.00	4744541.67	0	0
4485271	119348.25	85034.75	6796.88	0	0
303152	172108.06	0.00	167806.86	0	0
2242873	325172.83	0.00	1033505.62	0	0
160322	589512.29	0.00	656207.64	0	1

Sample: 4

	amount	newbalanceOrig	oldbalanceDest	isFraud	type_encoded
4980395	145471.80	0.0	536642.94	0	0
829923	229729.59	0.0	2509721.61	0	0
1139060	307204.15	0.0	0.00	0	0
2442880	132028.95	0.0	235602.00	0	0
3801355	86062.95	0.0	173417.89	0	1

Sample: 5

	amount	newbalanceOrig	oldbalanceDest	isFraud	type_encoded
4482144	422226.21	0.0	169806.64	0	0
4823144	89819.42	0.0	97041.88	0	0
5317836	28427.20	487775.8	5280197.72	0	0
2594531	141556.54	0.0	111198.42	0	1
4131025	122244.60	0.0	462268.60	0	0

Sample: 6

	amount	newbalanceOrig	oldbalanceDest	isFraud	type_encoded
4183130	578827.96	0.00	10000000.00	0	1
629782	286765.53	0.00	1745372.47	0	0
5355506	178724.31	0.00	2911406.52	0	0
5655443	148660.57	20239.43	1665.26	0	0
2504500	425978.49	0.00	0.00	0	1

Sample: 7

	amount	newbalanceOrig	oldbalanceDest	isFraud	type_encoded
2050590	442649.90	0.0	694640.80	0	1
4941732	2887664.69	0.0	1219344.66	0	1
2537931	204008.30	0.0	715695.96	0	0
4884430	288590.60	0.0	432862.95	0	0
3386037	259561.19	0.0	280200.42	0	0

Sample: 8

	amount	newbalanceOrig	oldbalanceDest	isFraud	type_encoded
2156911	42337.63	0.0	3039631.26	0	1
3902369	483099.37	0.0	863165.68	0	0
4506853	218076.98	0.0	214869.52	0	1
2589081	264503.99	0.0	3194404.18	0	1
5647338	54712.05	0.0	753096.55	0	0

Sample: 9

	amount	newbalanceOrig	oldbalanceDest	isFraud	type_encoded
1290127	210378.25	0.0	409457.82	0	0
5950890	144001.64	0.0	1311.14	0	0
3646751	528087.25	0.0	625616.91	0	1
15360	23850.09	0.0	1270603.21	0	0
1723129	874624.59	0.0	3064342.13	0	1

Sample: 10

	amount	newbalanceOrig	oldbalanceDest	isFraud	type_encoded
4259817	375309.70	0.00	3933314.32	0	0
4902596	14545.10	152144.90	0.00	0	0
1432642	20359.16	4865.84	0.00	0	0
4779699	66907.75	0.00	647254.60	0	0
2408360	53254.41	0.00	0.00	0	0

Sample: 11

	amount	newbalanceOrig	oldbalanceDest	isFraud	type_encoded
2120530	66282.23	6342.77	1014483.86	0	0
1697184	333177.67	0.00	0.00	0	0
3470824	140535.20	0.00	1744574.00	0	0
73408	267305.54	0.00	621246.77	0	0
2515515	360353.86	0.00	0.00	0	0

Sample: 12

	amount	newbalanceOrig	oldbalanceDest	isFraud	type_encoded
974709	48870.06	0.00	397367.81	0	0
3850600	265903.34	0.00	427882.59	0	0
5262619	446908.85	0.00	0.00	0	1
240051	290.44	102938.56	434193.07	0	0
3571690	91786.75	0.00	517538.10	0	0

Sample: 13

	amount	newbalanceOrig	oldbalanceDest	isFraud	type_encoded
1012802	49049.40	0.0	1287610.02	0	0
2676136	375164.75	0.0	967938.87	0	0
2929001	1748206.08	0.0	3009038.55	0	1
5692417	55891.25	0.0	652060.18	0	0
1624764	195846.12	0.0	236539.55	0	0

Sample: 14

	amount	newbalanceOrig	oldbalanceDest	isFraud	type_encoded
3649386	56357.42	0.0	176039.37	0	0
3835067	226954.68	0.0	66159.33	0	0
481000	333585.71	0.0	623753.42	0	0
4999124	246474.04	0.0	2069474.29	0	0
5759118	101547.95	0.0	784968.56	0	0

Sample: 15

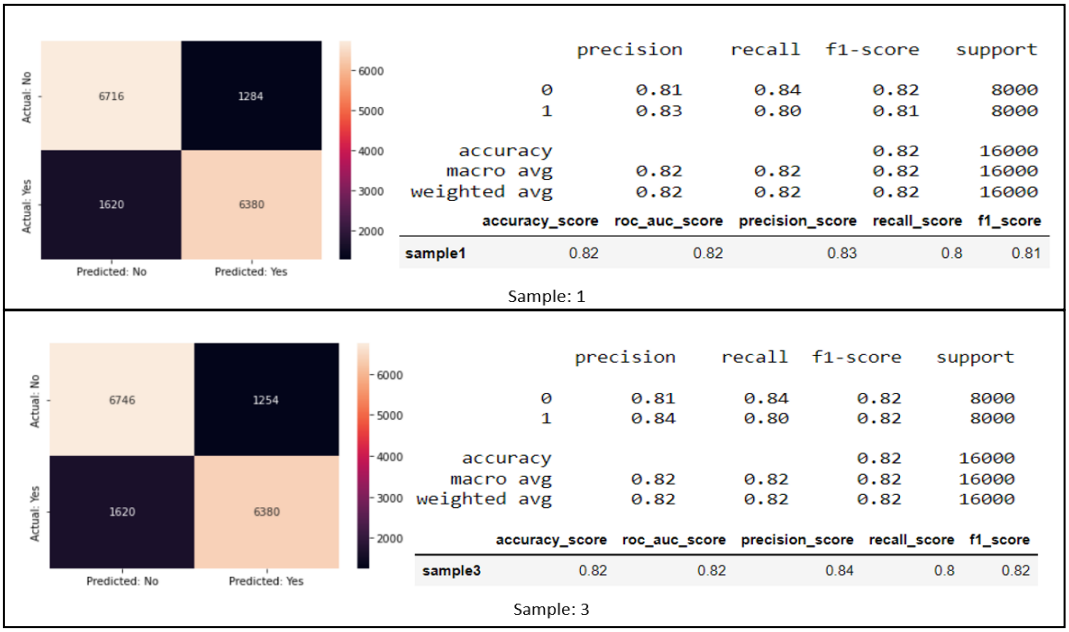
	amount	newbalanceOrig	oldbalanceDest	isFraud	type_encoded
214277	6292.79	0.00	899401.71	0	0
5392157	162401.90	0.00	0.00	0	0
1614378	188506.75	91993.25	102931.30	0	0
4591566	10354.26	0.00	1336858.22	0	0
5207428	59014.83	11705.17	400206.43	0	0

Fig 9: Sampling

7. Modelling and Evaluation:

As mentioned above totally 15 samples been taken from the dataset with equally distributed fraud and non-fraud. From 15 samples, 12 samples were considered for the training and 3 samples were considered for the testing. Ensemble technique with ADA boost classifier has been applied on all the 15 samples. Below shown figure is sample 1 and sample 3 and prediction has been done for training set.

From the figure we can see that the accuracy of the score is 80% and the accuracy for



	precision	recall	f1-score	support
0	0.81	0.84	0.82	8000
1	0.83	0.80	0.81	8000
accuracy			0.82	16000
macro avg	0.82	0.82	0.82	16000
weighted avg	0.82	0.82	0.82	16000

	accuracy_score	roc_auc_score	precision_score	recall_score	f1_score
sample1	0.82	0.82	0.83	0.8	0.81

Sample: 1

	precision	recall	f1-score	support
0	0.81	0.84	0.82	8000
1	0.84	0.80	0.82	8000
accuracy			0.82	16000
macro avg	0.82	0.82	0.82	16000
weighted avg	0.82	0.82	0.82	16000

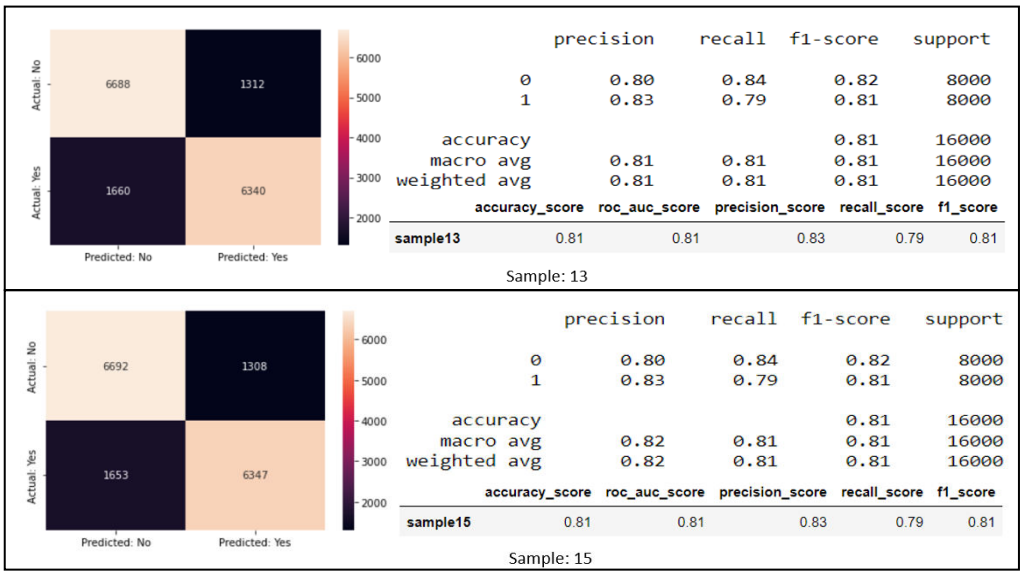
	accuracy_score	roc_auc_score	precision_score	recall_score	f1_score
sample3	0.82	0.82	0.84	0.8	0.82

Sample: 3

all the samples will be discussed below.

Fig 10: Results for Training sample

Similarly, testing has been performed on the remaining three sample to check if the performance metrics matches the same as of training sample. From the test sample from the below figure, we can say that the accuracy is more or less similar. So, we can proceed further with the assumption that our data is balanced but to further verify and implement these modeling in real time we need to compare this with one of the classification ML algorithms and confirm the performance metrices.



	precision	recall	f1-score	support
0	0.80	0.84	0.82	8000
1	0.83	0.79	0.81	8000
accuracy			0.81	16000
macro avg	0.81	0.81	0.81	16000
weighted avg	0.81	0.81	0.81	16000

	accuracy_score	roc_auc_score	precision_score	recall_score	f1_score
sample13	0.81	0.81	0.83	0.79	0.81

Sample: 13

	precision	recall	f1-score	support
0	0.80	0.84	0.82	8000
1	0.83	0.79	0.81	8000
accuracy			0.81	16000
macro avg	0.82	0.81	0.81	16000
weighted avg	0.82	0.81	0.81	16000

	accuracy_score	roc_auc_score	precision_score	recall_score	f1_score
sample15	0.81	0.81	0.83	0.79	0.81

Sample: 15

Fig 11: Results for Test sample

After applying ada boost classifier report has been generated for all the 12 training samples and 3 test samples as shown below

	accuracy_score	roc_auc_score	precision_score	recall_score	f1_score
sample1	0.82	0.82	0.83	0.80	0.81
sample2	0.82	0.82	0.82	0.81	0.82
sample3	0.82	0.82	0.84	0.80	0.82
sample4	0.82	0.82	0.84	0.80	0.82
sample5	0.82	0.82	0.83	0.80	0.82
sample6	0.82	0.82	0.83	0.80	0.82
sample7	0.82	0.82	0.83	0.80	0.82
sample8	0.82	0.82	0.83	0.80	0.81
sample9	0.82	0.82	0.83	0.80	0.81
sample10	0.82	0.82	0.82	0.80	0.81
sample11	0.82	0.82	0.84	0.79	0.82
sample12	0.83	0.83	0.84	0.81	0.82

Fig 12. Training sample

	accuracy_score	roc_auc_score	precision_score	recall_score	f1_score
sample13	0.81	0.81	0.83	0.79	0.81
sample14	0.82	0.82	0.83	0.80	0.82
sample15	0.81	0.81	0.83	0.79	0.81

Fig 13: Test sample

		Accuracy Score	
		ADA Boost Classifer	Decsion Tree Classifier
Training sample	Sample 1	0.82	0.82
	Sample 2	0.82	0.82
	Sample 3	0.82	0.82
	Sample 4	0.82	0.82
	Sample 5	0.82	0.82
	Sample 6	0.82	0.82
	Sample 7	0.82	0.82
	Sample 8	0.82	0.82
	Sample 9	0.82	0.82
	Sample 10	0.82	0.81
	Sample 11	0.82	0.82
	Sample 12	0.83	0.83
Test Sample			
	Sample 13	0.81	0.81
	Sample 14	0.82	0.82
	Sample 15	0.81	0.81

Table 14: Accuracy Scores for Training & test samples

8. Conclusion:

As mentioned above, to confirm the accuracy and other performance metrices we have compared the ada boost ensemble classification technique with the decision tree classification. As shown in the below figure we can confirm that the modeling is providing the same accuracy based on comparison with DT classifier. Now we can implement this algorithm in real-time to detect the fraud happening during transaction with approximately 81 percent of accuracy.

1. E. A. Lopez-Rojas and S. Axelsson. “Multi Agent Based Simulation (MABS) of Financial Transactions for Anti Money Laundering (AML)”
2. Taiwo, O. A. (2010). Types of Machine Learning Algorithms, New Advances in Machine Learning, Yagang Zhang (Ed.), ISBN: 978-953-307-034-6, InTech, University of Portsmouth United Kingdom. Pp 3 – 31. Available at InTech open website: www.intechopen.com
3. Good, I.J. (1951). Probability and the Weighing of Evidence, Philosophy Volume 26, Issue 97, 1951. Published by Charles Griffin and Company, London 1950.Copyright © The Royal Institute of Philosophy 1951, pp. 163-164.
4. Kotsiantis, S. B. (2007). Supervised Machine Learning: A Review of Classification Techniques. Informatica 31 (2007). Pp. 249 – 268. Retrieved from IJS website: wen.ijs.si
5. Bishop, C. M. (1995). Neural Networks for Pattern Recognition. Clarendon Press, Oxford, England. 1995. Oxford University Press, Inc. New York, NY, USA ©1995 ISBN:0198538642 Available at: cs.du.edu.
6. Bishop, C. M. (1995). Neural Networks for Pattern Recognition. Clarendon Press, Oxford, England. 1995. Oxford University Press, Inc. New York, NY, USA ©1995 ISBN:0198538642 Available at: cs.du.edu
7. Tapas Kanungo, D. M. (2002). A local search approximation algorithm for k-means clustering. Proceedings of the eighteenth annual symposium on Computational geometry. Barcelona, Spain: ACM Press
8. Ye Ren; Le Zhang; P.N. Suganthan ; Ensemble Classification and Regression-Recent Developments, Applications and Future Directions, reteived from: ieeexplore.ieee.org