Stochastic Modelling and Computational Sciences

A MACHINE LEARNING BASED MODEL FOR EARLY DETECTION AND PREVENTION OF ONLINE FINANCIAL FRAUD TRANSACTIONS

Atul Sharma[1] and Dr. Mamta Bansal[2]
[1]Research Scholar and [2]Professor, Shobhit Institute of Engineering & Technology (Deemed-To-Be University), Meerut (U.P.)

ABSTRACT
As per the modern trend the usage of digitization in terms of financial transactions inward or outward has been gaining significant importance due to its ease and availability but at the same time the cases of financial malpractices have been on the rise globally. In order to keep a check on this menace this research project was undertaken as a part of curriculum. The objective of this research paper is to classify real-time online financial transactions into 'fraud' or 'not-fraud' based on machine learning methodology. We all know it very well that the basic definition of fraud analytics is about gathering and storing relevant data. This data is used for generating patterns, discrepancies and anomalies with the use of Data Mining techniques. The findings of the research are then translated into insights which can be used to avoid possible threats beforehand. The research process requires training a machine learning model in order to classify online transactions into fraudulent or non-fraudulent payments. A real-time dataset consisting of online Banking Transactions was analyzed to know what type of transactions were responsible for the fraud.

Highlight: The highlight of this research project lies in the fact that this unique project has been undertaken using a unique methodology that has been implemented using Python Language.

Keywords: Online payment fraud; machine learning; Python Programming

I. INTRODUCTION
Fraud Detection Using Machine learning deploys a machine learning (ML) model [1]. The traditional methods which involve manual interventions are time consuming, expensive and inaccurate when we talk of financial fraud detection. We are living in a world which is rapidly adopting digital payments systems. Credit card and payments companies are experiencing a very rapid growth in their transaction volume [2]. In third quarter of 2018, PayPal Inc (a San Jose based payments company) processed 143 billion USD in total payment volume [3]. The introduction of online payment systems has helped a lot in the ease of payments but at the same time, it increased in the number of payment frauds. Online payment frauds have been on the rise as the process of digitization has become popular with the time.. With this objective in mind this project was undertaken. We propose to execute an exhaustive research based on the data extracted from Kaggle using Machine Learning. We show that our proposed approaches are able to detect fraud transactions with high accuracy and reasonably low number of false positives [4].

The last section of this research deals in a brief conclusion, in addition to detailed future possibilities.

II. RELEVANT RESEARCH
Machine Learning based approaches involve ANN (Artificial Neural Networks), SVM (Support Vector machines), HMM (Hidden Markov Models), clustering etc. We can find that many such researches have been taken up earlier using different tools and methodologies [5].

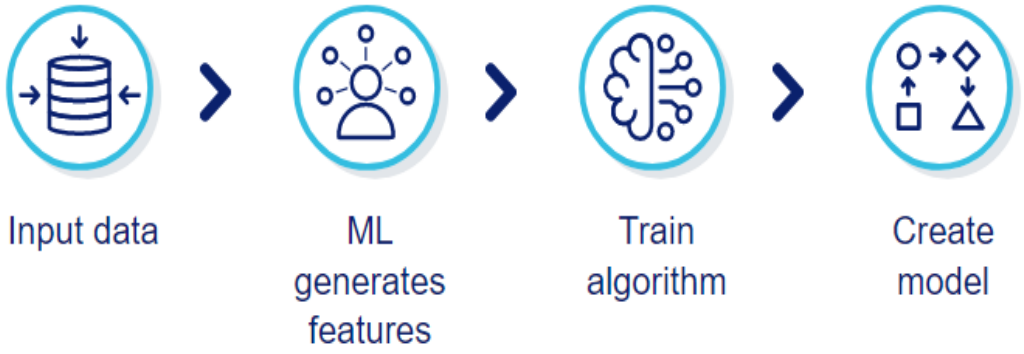

Figure 1: Working of Machine Learning system

III. DATASET AND ANALYSIS

In the recent past, Artificial Intelligence has been used as an effective tool which can successfully handle huge databases together with algorithms which may be complex but easy to interpret. The proposed framework offers the possibility of detecting and checking fiscal fraud which has been a prominent topic of research in the past. From the literature, data mining techniques present several possibilities for data processing with the aim at fraud analysis [6].

Machine learning models usually perform better than other predictive and non-predictive linear models when we talk about perfection and predictive capacity. These usually permit more accurate predictions and estimations as compared to other types of models [7].

The figure below shows the extracted version of data that has been mined from the dataset using Python language. We are using a non-simulated dataset which contains legitimate as well as fraud transactions from the duration 01st January 2020 to 31st December 2022.

Step	Type	Amount	NameOrig	OldbalanceOrg	NewbalanceOrig	NameDest	OldbalanceDest	NewbalanceDest	isFraud	isFlaggedFraud
1	PAYMENT	9839.64	C123100681	170136	160296.36	M197978715	0	0	0	0
1	PAYMENT	1864.28	C166654429	21249	19384.72	M204428222	0	0	0	0
1	TRANSFER	181	C130548614	181	0	C553264065	0	0	1	0
1	CASH_OUT	181	C840083671	181	0	C38997010	21182	0	1	0
1	PAYMENT	11668.14	C204853772	41554	29885.86	M123070170	0	0	0	0
1	PAYMENT	7817.71	C90045638	53860	46042.29	M573487274	0	0	0	0
1	PAYMENT	7107.77	C154988899	183195	176087.23	M408069119	0	0	0	0
1	PAYMENT	7861.64	C191285043	176087.23	168225.59	M633326333	0	0	0	0
1	PAYMENT	4024.36	C126501292	2671	0	M117693210	0	0	0	0
1	DEBIT	5337.77	C712410124	41720	36382.23	C195600860	41898	40348.79	0	0
1	DEBIT	9644.94	C190036674	4465	0	C997608398	10845	157982.12	0	0
1	PAYMENT	3099.97	C249177573	20771	17671.03	M209653912	0	0	0	0
1	PAYMENT	2560.74	C164823259	5070	2509.26	M972865270	0	0	0	0
1	PAYMENT	11633.76	C171693289	10127	0	M801569151	0	0	0	0
1	PAYMENT	4098.78	C102648383	503264	499165.22	M163537821	0	0	0	0
1	CASH_OUT	229133.9	C905080434	15325	0	C476402209	5083	51513.44	0	0
1	PAYMENT	1563.82	C761750706	450	0	M173121798	0	0	0	0
1	PAYMENT	1157.86	C123776263	21156	19998.14	M187706290	0	0	0	0
1	PAYMENT	671.64	C203352454	15123	14451.36	M473053293	0	0	0	0

Figure 2: Sample transaction sheet

Figure 2 above shows the partial layout of extracted data from the downloaded dataset for experimentation. The data has been extracted using Python command.

Sr.No	Field Name	Description	Type
01	Step	Period of time with 1 step equals to one hour of time.	Numeric
02	Type	Type of online transaction	Non-Numeric
03	Amount	Transaction amount in local currency	Numeric
04	NameOrig	Identification number of the payer	Alphanumeric
05	OldbalanceOrg	Initial balance amount of the payer	Numeric
06	NewbalanceOrig	Final balance of the payer after the transaction	Numeric
07	NameDest	Identification number of the payee	Alphanumeric
08	OldbalanceDest	Opening balance of the payee	Numeric
09	NewbalanceDest	The new balance of the payee after the transaction	Numeric
10	isFraud	Fraud transaction (Boolean)	Numeric

Table 1: Description of the fields used in the research project

IV. METHODOLOGY

The methodology consists of the various steps as mentioned in the steps as shown below.

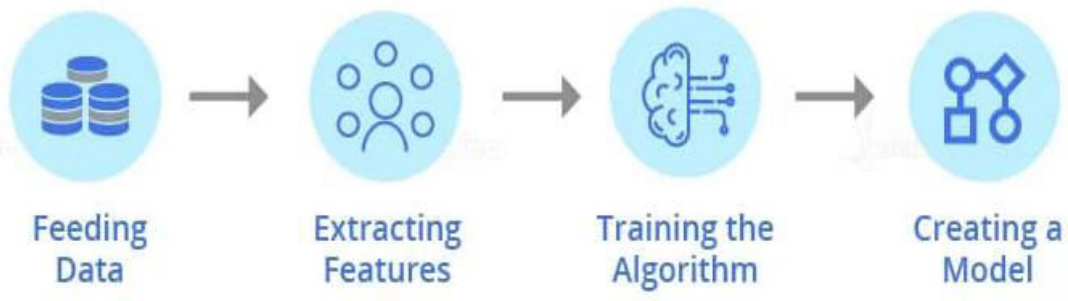

Figure 3: Outline of Fraud detection using Machine Learning

Steps used in the experimentation

Step 1: Firstly, we go ahead to import necessary Python libraries which are essential for any project like pandas and numpy to name a few as shown below:

- Import pandas as pd
- Import numpy as np

Step 2: Read the transaction file from the source as shown below:

data=pd.read_csv("C:\\Users\\user\\Desktop\\Test\\final.csv")

Step 3: Confirming the uploaded data using print command:

```
>>> print(data.head())
   step      type    amount  ... newbalanceDest  isFraud  isFlaggedFraud
0     1   PAYMENT   9839.64  ...            0.0        0               0
1     1   PAYMENT   1864.28  ...            0.0        0               0
2     1  TRANSFER    181.00  ...            0.0        1               0
3     1  CASH_OUT    181.00  ...            0.0        1               0
4     1   PAYMENT  11668.14  ...            0.0        0               0

[5 rows x 11 columns]
```

Step 4: Checking for the consistency of the data by finding out for any not null value in the dataset: -

```
>>> print(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
>>> print('Total number of null rows',data.isnull().values.any())
Total number of null rows False
```

The above command verifies that there is no null value in the dataset which indicates that the end results would be significant.

Step 5: Finding the count of transaction type in order to understand the frequency of test data: -

```
>>> print(data.type.value_counts())
type
CASH_OUT    373641
PAYMENT     353873
CASH_IN     227130
TRANSFER     86753
DEBIT         7178
Name: count, dtype: int64
```

Step 6: Graphical distribution of Transaction type for quick understanding of distribution of data

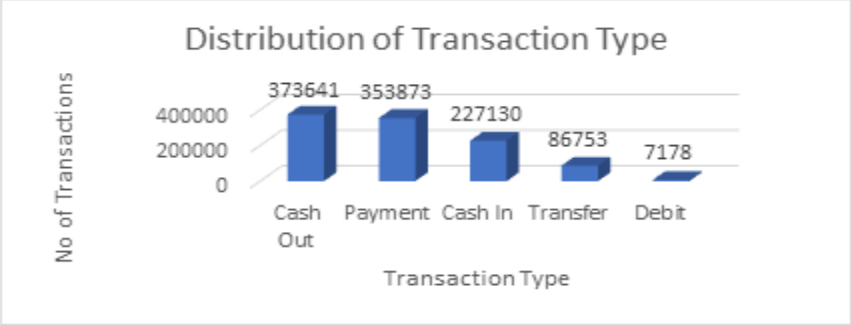

Figure 4: Distribution of Transaction Type

Step 7: Understanding the correlation among the variables present in the data with respect to "isFraud" column: -

```
>>> correlation=data.corr()
>>> print(correlation["isFraud"].sort_values(ascending=False))
isFraud           1.000000
amount            0.128862
step              0.045030
oldbalanceOrg     0.003829
newbalanceDest   -0.000495
oldbalanceDest   -0.007552
newbalanceOrig   -0.009438
isFlaggedFraud         NaN
Name: isFraud, dtype: float64
```

The results above show that there is a positive linear relationship between 'isFraud' and 'amount' fields.

- A correlation coefficient greater than zero indicates a positive relationship while a value less than zero signifies a negative relationship.
- A value of zero indicates no relationship between the two variables being compared [8].

Step 8: Transform isFraud column to "No Fraud" or "Fraud" values for better analysis of the test dataset: -

```
>>> data["isFraud"]=data["isFraud"].map({0:"No Fraud",1:"Fraud"})
>>> print(data.head())
   step      type    amount  ... newbalanceDest   isFraud  isFlaggedFraud
0     1   PAYMENT   9839.64  ...            0.0  No Fraud               0
1     1   PAYMENT   1864.28  ...            0.0  No Fraud               0
2     1  TRANSFER    181.00  ...            0.0     Fraud               0
3     1  CASH_OUT    181.00  ...            0.0     Fraud               0
4     1   PAYMENT  11668.14  ...            0.0  No Fraud               0

[5 rows x 11 columns]
```

Step 9: Transforming categorical features (type field) into numerical values since data can be analyzed in numerical format: -

```
>>> data["type"]=data["type"].map({"CASH_OUT":1,"PAYMENT":2,"CASH_IN":3,"TRANSFER":4,"DEBIT":5})
>>> print(data.head())
   step  type    amount  ... newbalanceDest  isFraud  isFlaggedFraud
0     1     2   9839.64  ...            0.0        0               0
1     1     2   1864.28  ...            0.0        0               0
2     1     4    181.00  ...            0.0        1               0
3     1     1    181.00  ...            0.0        1               0
4     1     2  11668.14  ...            0.0        0               0

[5 rows x 11 columns]
```

Online Payments Fraud Detection Model proposed by the researcher

Now we go ahead to split our testing data into training and testing data sets which would identify transactions as 'fraud' or 'not fraud' based on the developed model.

```
>>> from sklearn.model_selection import train_test_split
>>> x=np.array(data[["type","amount","oldbalanceOrg","newbalanceOrig"]])
>>> y=np.array(data[["isFraud"]])
```

Going further we would train the online payments fraud detection model as shown below

```
>>> from sklearn.model_selection import train_test_split
>>> x=np.array(data[["type","amount","oldbalanceOrg","newbalanceOrig"]])
>>> y=np.array(data[["isFraud"]])
>>> from sklearn.tree import DecisionTreeClassifier
>>> xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.10,random_state=42)
>>> model=DecisionTreeClassifier()
>>> model.fit(xtrain,ytrain)
DecisionTreeClassifier()
>>> print(model.score(xtest,ytest))
0.9993801140590132
```

It is clear that the model score achieved is 99.93% success rate which is a great model performance.

Finally, we go ahead to evaluate the success rate of the proposed model by replacing the variables with data in order to classify any transaction as "Fraud" or "Not Fraud"

```
>>> from sklearn.model_selection import train_test_split
>>> x=np.array(data[["type","amount","oldbalanceOrg","newbalanceOrig"]])
>>> y=np.array(data[["isFraud"]])
>>> from sklearn.tree import DecisionTreeClassifier
>>> xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.10,random_state=42)
>>> model=DecisionTreeClassifier()
>>> model.fit(xtrain,ytrain)
DecisionTreeClassifier()
>>> print(model.score(xtest,ytest))
0.9993896507657976
>>> features=np.array([[4,9000.60,9000.60,0.0]])
>>> print(model.predict(features))
['Fraud']
```

Henceforth the proposed model is verified as successful based on the data analysis with the help of machine learning model proposed by the researcher.

V. CONCLUSION AND FUTURE WORK

In this research paper, a unique research model has been proposed and verified using historical statistical data & Python language. We have talked about various scenarios of fraudulent actions online in the introduction section discovered the whole process of fraud detection using Python and machine learning. The basic feature of this model is to classify the given dataset transactions as a fraudulent or genuine transaction. With the given dataset, this model has proved to result in better forecasting. The dataset is preprocessed along with the feature selections, the data is then sent to classification through Python interface into various factors before letting it to final processed information. The final output is to obtain the transactions as true or fraudulent. This model can be then tested and trained with the larger data volume in future, so as to get more precise and accurate results. The model can also be upgraded to test dynamic data in future for more advanced research. As the next step in this research program, the focus will be upon the implementation of a 'suspicious' scorecard on a real data-set and its evaluation. The main tasks will be to build scoring models to predict fraudulent behavior, taking into account the fields of behavior that relate to the different types of credit card fraud identified in this paper, and to evaluate the associated ethical implications. The plan is to take one of the European countries, probably Germany, and then to extend the research to other EU countries [9].

APPENDIX

https://github.com/shindenikhil659/Online-Payments-Fraud-Detection-with-Machine-Learning

VI. ACKNOWLEDGEMENT

My sincere thanks to Dr. Mamta Bansal for her valuable insights towards the development of this predictive model.

REFERENCES

[1]. Ahmed, M. H. (2023). A Review: Credit Card Fraud Detection in Banks using Machine Learning Algorithms.ScienceOpen.http://dx.doi.org/10.14293/s2199-1006.1.sor-.ppfi7p0.v2

[2]. Slotter, K. (1997). Plastic payments: Trends in credit card fraud. *PsycEXTRA Dataset*. https://doi.org/10.1037/e317072004-001

[3] *PayPal Newsroom*. (n.d.). PayPal Newsroom. Retrieved May 9, 2023, from https://www.paypal.com/stories/us/paypalreports-third-quarter-2018-results

[4]. Oza, Aditya. "Fraud Detection Using Machine Learning - Stanford University." Https://Cs229.Stanford.Edu/Proj2018/Report/261.Pdf, cs229.stanford.edu/proj2018/report/261.pdf.

[5]. R, V. C., Asha, V., Prasad, A., Das, S., Kumar, S., & P, S. S. (2023, February 23). Support Vector Machine (SVM) and Artificial Neural Networks (ANN) based Chronic Kidney Disease Prediction. *2023 7th International Conference on Computing Methodologies and Communication (ICCMC)*. http://dx.doi.org/10.1109/iccmc56507.2023.10083622

[6]. Sarker. (2021). Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science*, *2*(3), 1–21. https://doi.org/10.1007/s42979-021-00592

[7]. Building predictive models using penalized linear methods. (2015). In Machine Learning in Python® (pp. 165–210). John Wiley & Sons, Inc. http://dx.doi.org/10.1002/9781119183600.ch5

[8]. Probability that each growth coefficient is greater or less than zero for each of the five species analyzed here. (n.d.). https://doi.org/10.7717/peerj.3102/table-4

[9]. Difficulties experienced when making web sales to other EU countries, 2016. (2019b). https://doi.org/10.1787/65fe051b-en