Remittances Review
January, 2024
Volume: 9, No: 1, pp. 1897-1914
ISSN: 2059-6588(Print) | ISSN 2059-6596(Online)

Received: 5 January 2024, Accepted: 10 February 2024

DOI: https://doi.org/10.33282/rr.vx9il.82

SUSTAINABLE FRAUD DETECTION IN GREEN FINANCE EMPOWERED WITH MACHINE LEARNING APPROACH

Danial Jamil[*1], Ahmed Iqbal[2], Bijaya Bhattarai[3], Md Mahbubur Rahman[4], Dr.E.N.Ganesh[5], Muhammad Asim[6], Tariq Rafique[7], M Asjad Abbasi[8]

[1]Research Scholar, Department of Computer Science and IT, Ghazi University Dera Ghazi Khan, Pakistan, Email: Danialjamil05@gmail.com

[2]Design Officer, Aircraft Division (ADIC), National Aerospace Science and Technology Park (NASTP), Pakistan, Email: ahmaddar1822@gmail.com

[3]University of Tasmania, Australia, Email: beezai939@gmail.com

[4]University of Chittagong, Bangladesh, Email: mahbubur.merchandiser@gmail.com

[5]Research and Development, Department of ECE, SPIHER Chennai, India, Email: enganesh50@gmail.com

[6]M.phil Scholar, Artificial Intelligence, The Islamia University Bahawalpur, Pakistan, Email: muhammad.aasim2024@gmail.com

[7]Assistant Professor Dadabhoy Institute of Higher Education, Karachi, Pakistan, Email: dr.tariq1106@gmail.com

[8]Department of Management Science, Preston University, kohat Islamabad Campus, Islamabad, Pakistan, Email: asjad.abbasi@gmail.com

*Corresponding author

Abstract—Prior to machine learning, businesses would employ a rule-based strategy to identify fraud by examining recurring and obvious indicators. Countless fraud scenarios are executed by pure rule-based algorithms, which are personally crafted by an individual. The goal of the paper is to create precise deep learning and machine learning models for the Green Finance fraud detection. Fraud in real-time transactions cannot be detected by conventional rule-based methods. The study tackles the problem of unbalanced data by using the PaySim dataset, which replicates mobile transactions. The performance of a number of algorithms is assessed, including Random Forests, Recurrent Neural Networks, and K-Nearest Neighbors. To uncover hidden patterns in user transactions, the application of long short-term memory models and artificial neural networks is investigated. The study talks about difficulties including data cleaning and tweaking hyperparameters. The results support the development of more precise and effective fraud detection systems, which helps Green Fiannce lower losses and preserve consumer confidence.

Keywords: Classification of fraud versus non-fraud, Grouping Identification of anomalies, Amount of transaction, Frequency of transactions.

Remittances Review
January, 2024
Volume: 9, No: 1, pp. 1897-1914
ISSN: 2059-6588(Print) | ISSN 2059-6596(Online)

INTRODUCTION

A illegal deceit performed by someone acting dishonestly and falsely is called fraud. UK Finance (2019). reports that in 2018, unlawful financial fraud losses using checks, payment cards, and remote banking were £844.8 million, representing a 16 per cent increase over 2017. Because of COVID-19 and the widespread implementation of lockdowns, internet transactions are now even more popular.(Alarfaj et al., 2022).

Prior to machine learning, businesses would employ a rule-based strategy to identify fraud by examining recurring and obvious indicators. Countless fraud scenarios are executed by pure rule-based algorithms, which are personally crafted by an individual. Rule-based scenarios now days need to be adjusted much too frequently to stay current with security patches. Furthermore, rule-based systems are ineffective when it comes to streaming real-time data, which is important for green fianance.(Cui, Yan, & Wang, 2021) .

However, machine learning may be used to stream data in real time, identify hidden patterns in user behavior, and determine the likelihood of fraudulent activity. In comparison to rule-based algorithms,

step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
1	PAYMENT	9839.64	C1231006815	170136.00	160296.36	M1979787155	0.00	0.00	0	0
1	PAYMENT	1864.28	C1666544295	21249.00	19384.72	M2044282225	0.00	0.00	0	0
1	TRANSFER	181.00	C1305486145	181.00	0.00	C553264065	0.00	0.00	1	0
1	CASH_OUT	181.00	C840083671	181.00	0.00	C38997010	21182.00	0.00	1	0
1	PAYMENT	11668.14	C2048537720	41554.00	29885.86	M1230701703	0.00	0.00	0	0
...	...	...	...	...	...	...	...	...	...	...
743	CASH_OUT	339682.13	C786484425	339682.13	0.00	C776919290	0.00	339682.13	1	0
743	TRANSFER	6311409.28	C1529008245	6311409.28	0.00	C1881841831	0.00	0.00	1	0
743	CASH_OUT	6311409.28	C1162922333	6311409.28	0.00	C1365125890	68488.84	6379898.11	1	0
743	TRANSFER	850002.52	C1685995037	850002.52	0.00	C2080388513	0.00	0.00	1	0
743	CASH_OUT	850002.52	C1280323807	850002.52	0.00	C873221189	6510099.11	7360101.63	1	0

machine learning also requires less time. This research aims to create accurate deep learning and machine learning models that can compare and prevent fraudulent transactions. To achieve the task, I must: Obtain a financial dataset. Eliminate superfluous data fields from the dataset. Identify appropriate metrics to assess the model's performance. After testing a number of models, identify the most effective and continue to refine the less successful models. (Pumsirirat & Liu, 2018). Unfortunately, because of personal data and General Data Protection Regulations, there aren't many publicly available datasets for fraud detection. The PaySim simulator created the artificial dataset known as PaySim. It creates a dataset that resembles a normal one using a group of private datasets. PaySim is a mobile transaction simulator that uses real-world transaction samples. These transactions were taken from several financial logs from an African mobile money provider over a period of one month. A global corporation provided the logs, which. offers mobile financial services in

over 14 countries throughout the globe. The dataset, which comprises 11 columns and 6362620 rows, was scaled down to one-fourth of its original size (Chaudhary, Yadav, & Mallick, 2012).

Figure 1: The dataset's column data is shown.

It will be necessary to adjust this severely imbalanced dataset for the models to produce accurate predictions. Just 0.13% of transactions are fraudulent, whereas 99.87% of transactions are legitimate.

```
pos, neg = np.bincount(df.isFraud)
total = neg + pos
print("Total transactions:\t\t", total,"\n")
print("Genuine transactions:\t\t", pos, "(", round((pos * 100 / total), 2), "%)\n")
print("Fraudulent transactions:\t", neg, "(", round((neg * 100 / total), 2), "%)\n")
```

```
Total transactions:             6362620

Genuine transactions:           6354407 ( 99.87 %)

Fraudulent transactions:        8213 ( 0.13 %)
```

Figure 2: Amount of genuine and fraudulent transactions

FLOW

A. Steps

A step connects a time interval to the actual world. One step in this dataset corresponds to one hour. It's a 30-day simulation with 743 steps in total.

```
print("Steps - from {} to {}.".format(df['step'].min(), df['step'].max()))
```

```
Steps - from 1 to 743.
```

Figure 3: Total number of steps

B. Type

```
# Display the different types of transactions of the TYPE field
df['type'].unique()
```

```
array(['PAYMENT', 'TRANSFER', 'CASH_OUT', 'DEBIT', 'CASH_IN'],
      dtype=object)
```

Figure 4: Different values in the 'type' column

There exist five distinct categories of transactions:

Payment is the phrase used to describe a transaction in which a client pays a merchant to get products or

January, 2024
Volume: 9, No: 1, pp. 1897-1914
ISSN: 2059-6588(Print) | ISSN 2059-6596(Online)

services. The recipient's account balance rises (i.e., money is credited to his account)., while the sender's account balance falls. TRANSFER: A transaction is referred to as a transfer when money is sent between users using a mobile money service platform. CASH_OUT - The merchant acts as an ATM for the clients, allowing them to take out cash to lower their account balance.

CASH_IN: The merchant acts as an ATM for the consumers, allowing them to pay the merchants with cash to boost the amount of their accounts.

DEBIT: A transaction is referred to as a debit when a consumer transfers funds from a mobile money service to a bank account. The account balance is decreased in the same way as a CASH_OUT transaction.

C. Amount

The amount in local currency that was transacted.

```
df['nameOrig'].unique()
```

```
array(['C1231006815', 'C1666544295', 'C1305486145', ..., 'C1162922333',
       'C1685995037', 'C1280323807'], dtype=object)
```

Figure 5: Variations in the 'amount' column values

Literature Review

Green Finance use both supervised and unsupervised learning techniques to find fraud in their data. Supervised learning is becoming prevalent in several academic fields. To produce predictions, supervised learning needs a tagged dataset. On the other hand, unsupervised learning picks up on trends and irregularities to determine whether or not a transfer is fake. Deep learning for identifying fraud is evaluating and learning from a consumer's patterns and behaviors to determine if a transaction is authentic or fraudulent.(Pumsirirat & Liu, 2018). SVM is a solid, quick method that can handle small datasets. In this scenario, it is utilized to determine if a transaction is fraudulent or not, maximizing the margin between distinct classes. When there are nearly no fraud cases in an imbalanced dataset for fraud detection, Random Forests perform well (Chaudhary et al., 2012).

Depending on the user's behavior level, LSTM models may be used to assess whether a transaction is fraudulent or authentic by slicing through the data rather than only focusing on the transaction itself. Important data is never lost using this way. LSTM solves the vanishing gradient problem that vanilla RNNs face. The RNN weights in the preceding layers are impacted by the vanishing gradient issue. The gradient of the loss function approaches 0 when layers employ activation functions. This increases the difficulty of

training the network. However, the LSTM models are more challenging to (Dileep, Navaneeth, & Abhishek, 2021). Since they must do extra calculations to decide whether to keep or reject the data, they are more difficult to install and take longer to train. But LSTMs outperform conventional machine learning algorithms in terms of accuracy (Gupta et al., 2023).

The accuracy, sensitivity, specificity, and precision of logistic regression, KNN, and Naïve Bays are examined by Awoyemi, (Awoyemi, Adetunmbi, & Oluwadare, 2017). Logistic Regression was out to be the least effective of the three methods. But instead of using the PaySim dataset, it concentrates on accuracy, sensitivity, specificity, and precision. It seems sense to start by learning which algorithms may serve as the basis for fraud detection.(Al-Hashedi & Magalingam, 2021).

To fully eliminate fraud detection (Bandyopadhyay, Thakkar, Mukherjee, & Dutta, 2021) suggest employing a Stacked-RNN with 12 layers for the PaySim dataset. The RNN's accuracy is 99.87%, F1-score is 0.99 and the Mean Squared Error is 0.01. This research is an excellent illustration of an RNN architecture, even if Bandyopadhyay and Dutta utilise accuracy, F1-score, and mean squared error to assess the model's performance.

Accuracy is a terrible statistic in fraud detection because of the accuracy paradox. Additionally, the F1 score has a True Positive value ratio of 99%, which is the greatest figure since genuine transactions consistently outweigh fraudulent ones, which account for less than 1% of the dataset. Moreover, regression models are the main applications for mean squared error rather than categorization (Lakshmi & Kavilla, 2018).

When a model misclassifies, MSE does not penalize it as much as it might (Alarfaj et al., 2022).

In order to ascertain if a transaction is authentic or fraudulent, (Kaur, Pannu, & Malhi, 2019) suggests utilizing the algorithms Logistic Regression, Naïve Bayes, Random Forest, and XGBoost. They compare the variations using two distinct data segmentation techniques, test-train split and K-folds. Kaur further contrasts the various algorithms by their correctness from their study. The maximum accuracy is achieved by XGBoost, which is 99.95% (train-test split). and 96.46% (K-fold). In addition, Kaur recommends doing random and grid searches to identify the optimal parameters for a given method. Because the research provides the optimal settings for the recommended algorithms and makes use of the PaySim dataset, it's an excellent place to start. Nevertheless, it ignores relevant measures like the percentage of fraud that has gone unnoticed or the False Negative Rate.

Höppner, Baesens, Verbeke, and Verdonck (2022) analyze various metrics depending on the dataset itself and the results of applying the SMOTE method to produce artificially generated fraudulent transactions. Furthermore, they offer precise custom metrics—like the financial cost of a fraudulent transaction (assuming fraud is committed)., the financial cost of a legitimate transaction that is suspected of being fraudulent, and

different weights for values that are true negative, false negative, true positive, and false positive—to assess an algorithm's efficacy. This study provides a useful foundation for thinking about appropriate measures to take into account when comparing various models. They do not, however, make use of the PaySim dataset and instead provide more precise information, including actual statistics and admissions charges.

If a transaction is questionable, Nordling (2020). advises employing decision trees, random forests, and autoencoders to ascertain its authenticity. The Synthetic Minority Oversampling Technique is presented to overcome the unbalanced dataset by increasing the number of transactions of the less dominant class until the number of data points for both classes is equal. On the other hand, the study used a different dataset with AUROC, accuracy, and recall as measurements.(Alarfaj et al., 2022).

SVM is suggested by Pambudi, Hidayah, and Fauziati as a means of identifying authentic or fraudulent transactions. They evaluate the performance of several kernels with varying gammas and C values using the following metrics: precision, recall, F1-score, and Area Under Precision-Recall Curve (AUPRC). However, SVM calculation takes a long time, especially for large datasets with lots of columns. A model with the same set of parameters may provide completely different findings since the research does not address how their data is cleansed.

In general, several research suggest using XGBoost, Random Forest, SVM Logistic Regression, Recurrent Neural Network, and Naïve Bayes using measures like AUROC, Accuracy, and Recall. Research utilizing customized datasets suggest customized measures that more accurately determine if a model is operating well or poorly (Olivas, Guerrero, Martinez-Sober, Magdalena-Benedito, & Serrano, 2009)

METHODOLOGY

Measures

The confusion matrix has to be described before we can talk about the metrics that will be used:

True Positive (TP).: The number of authentic transactions that were accurately identified as authentic.

False Positive (FP).: The number of legitimate transactions that were mistakenly identified as fraudulent. This demonstrates the number of legitimate transactions that the model rejected because it believed them to be fraudulent. The most significant measure is False Negative (FN). It shows the number of fraudulent transactions that were assumed to be real. It shows that fraud has happened and has not been identified(Torres Berru, López Batista, Torres-Carrión, & Jimenez, 2020). The number of fraud transactions that were accurately identified as fraudulent (True Negative, TN). In conventional binary classification, measures like accuracy, AUC, F1-score, and so on must be maximized while the loss function must be minimized.

January, 2024
Volume: 9, No: 1, pp. 1897-1914
ISSN: 2059-6588(Print) | ISSN 2059-6596(Online)

One measure that is utilized to predict the accuracy of our model is accuracy. Unfortunately, since there are 99% legitimate transactions and fewer than 1% fraudulent transactions in this fraud detection dataset, accuracy will not be effective. This suggests that for any model, the accuracy will be 99% accurate. The accuracy dilemma thus arises.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Since F1-score emphasizes True Positive values (real transactions). through a mix of accuracy and recall, it would not function as effectively.

$$\text{Percision} = \frac{TP}{TP + FP}$$

$$\text{Recall(Sensitivity or True Positive Rate).} = \frac{TP}{TP + FN}$$

$$\text{Specificity(True Negative Rate).} = \frac{TN}{TN + FP}$$

$$\text{F1 Score} = 2 \times \frac{\text{Percision} * \text{Recall}}{\text{Percision} + \text{Recall}}$$

The AUC score would be useless since it only shows the possibility that a selected positive example will have a higher projected likelihood of being positive than negative.

The AUC score in fraud detection indicates the likelihood that a transaction is authentic as opposed to fraudulent. Because fraudulent transactions are insufficient, datasets are typically highly imbalanced, which results in every model receiving at least a 0.99 score or even a 1(Torres Berru et al., 2020).

Since it indicates the potential that a chosen positive example will have a higher anticipated chance of being positive than negative, the AUC score would not be helpful(Olivas et al., 2009).

The AUC score in fraud detection indicates the likelihood that a transaction is authentic as opposed to fraudulent. Because fraudulent transactions are insufficient, datasets are typically highly imbalanced, which results in every model receiving at least a 0.99 score or even a 1.

Remittances Review
January, 2024
Volume: 9, No: 1, pp. 1897-1914
ISSN: 2059-6588(Print) | ISSN 2059-6596(Online)

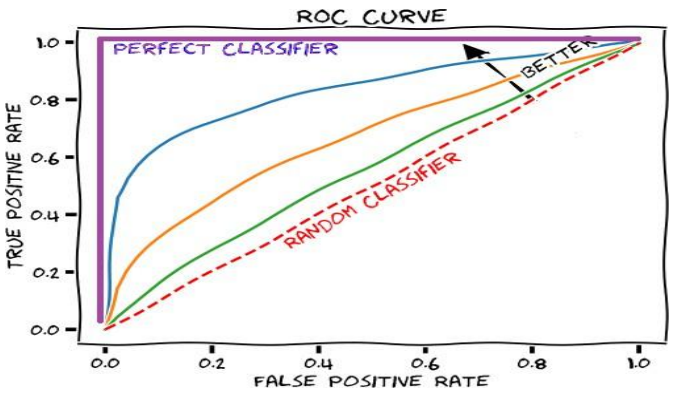


Figure 6: An AUC curve example

Consequently, better classification performance is indicated by a larger AUC. AUC (and ROC curves). may be overly optimistic when working with severely unbalanced data, as is the situation with fraud detection. An illuminating perspective of a classifier's performance is instead offered by the Area under the Precision-Recall Curve (AUPRC). For datasets that are imbalanced, meaning that one class—usually the positive one—dominates the other, AUPRC is a useful statistic. It is calculated to find the area under the PR curve, which shows the degree of agreement between recall and precision. This curve concludes in the bottom right corner, when recall and precision are both one, and starts in the top left corner, where recall and accuracy are both zero. The accuracy and recall of the various thresholds are calculated to retrieve the data between the start and finish points.

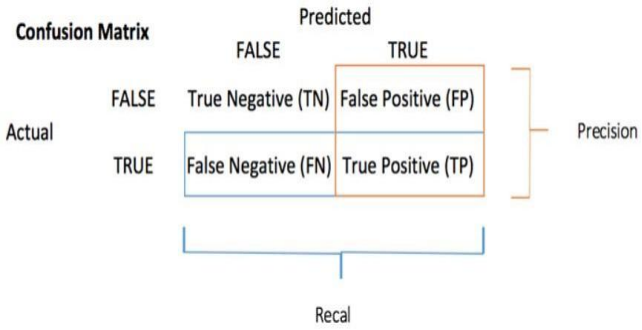


Figure 7: An example of a confusion matrix that illustrates how recall and accuracy are produced

Still, the percentage of fraud that the model detects is the most significant measure. This statistic is centered on the quantity of transactions that are False Negative. False Negatives are important because they show that fraudulent transactions are accepted as real (Karthika & Senthilselvi, 2023). As a result, Green Finance incur

losses. False Positives, or legitimate transactions mistakenly believed to be fraudulent, are less important since the bank must get in touch with the consumer to confirm them, or the customer can phone the bank to confirm that the "suspicious" transaction is being made. The following formula is used to compute the metric:

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Cases}}$$

$$\text{Missed Detection Rate} = 1 - \text{Accuracy}$$

$$\text{Missed Detection Rate} = \frac{\text{Total fraud}}{FN+TN}$$

D. Imbalanced Data Techniques

As was already indicated, in order for the models to produce accurate predictions, the PaySim dataset will need to be adjusted due to its extreme imbalance. Just 0.13% of transactions are fraudulent, whereas 99.87% of transactions are legitimate. When there is an imbalance in the representation of the classes, it is referred to as imbalanced data. When there is an imbalance, the model may entirely overlook the minority class, producing erroneous findings. For the model to produce correct predictions, the data must either be under- or oversampled, or the class weights must be adjusted.

E. Random Undersampling

Using random undersampling, samples from most of the class are chosen at random and removed. Until the dominant class has the same quantity of data as the minority class, samples are eliminated. This implies that most of the legitimate transactions will be eliminated from the PaySim dataset unless there are an equal number of fraudulent and legitimate transactions. Due to data loss, the models will have to work harder to learn, which will lower their performance. This may be quite troublesome. This implies that 99 percent of the information will be lost.

F. ClassWeights

Generally speaking, machine learning algorithms do not account for an unbalanced dataset, which means that incorrectly categorising a valid and incorrect example would result in the same penalty. But in unbalanced datasets, the penalty for incorrectly categorising a minority class must be greater than the penalty for incorrectly classifying a valid case (Torres Berru et al., 2020). The class weight is turned off by default when the algorithm is implemented, assigning equal weights to the two classes. The following formula is used to determine the weights when the class weight is set to balance:

$$\text{Formula} = \frac{n\_samples}{}$$

n_classes * n_samplesj

- Each class's weight is denoted by wj, where j is the class.
- Sn_samples indicates the total number of samples in the dataset.
- The variable n_classes indicates the total number of unique classes in the dataset. The number of lessons in the selected class is denoted by j.

Data purification

The dataset has to be enhanced or in an acceptable stage in order to get exact findings. The dataset will be much better if noise, missing values, and inconsistent fields are eliminated (Karthika & Senthilselvi, 2023). There are 11 columns and 6362620 rows in the dataset. To ensure accurate data cleaning, the following inquiries must be addressed:

Which are the fraudulent transactions?

There is 0.13% of transactions that involve fraud. Only transactions with the TRANSACTION type of TRANSFER or CASH_OUT are susceptible to fraud. We may observe from the second graph that TRANSFERs occur four times less frequently than CASH_OUTs. Thus, CASH_IN, DEBIT, and PAYMENT.

Transaction types that do not exhibit fraudulent conduct are eliminated since they are superfluous.

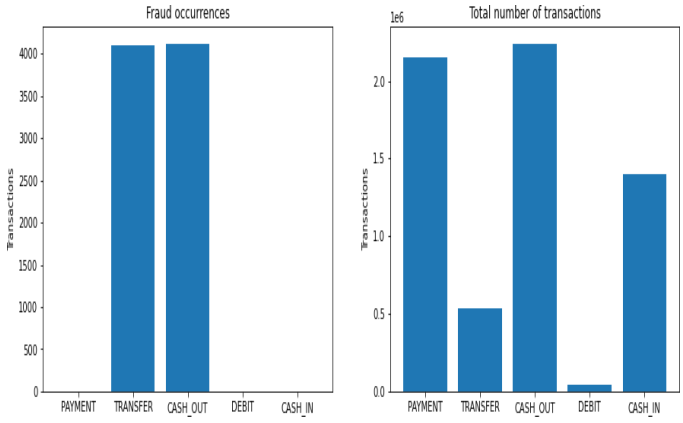


Figure 8: Transactions with fraud and a total number of transactions displayed

II. MODELS

A. Model 1

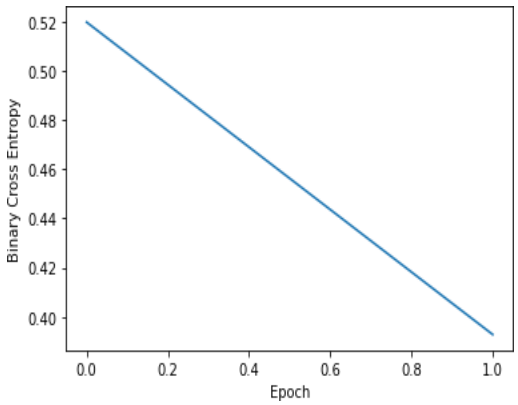
This model is based on a 12-layer recurrent neural network proposed in an article by (Bandyopadhyay et al., 2021)Nevertheless, they gauge the model's performance using MSE, accuracy, and F1-Score. I'll be

employing average precision-recall score, percent of fraud identified, and binary cross entropy in this research. The first model that has been suggested is:

Layer	**Output Shape**	**Parameters**	**Activation Function**
NormalRNN	(NotAny; 10; 128).	16640	Sigmoid
Drop out	0.2	0	NotAny
NormalRNN	(NotAny; 10; 64).	12352	Sigmoid
Drop out	0.2	0	NotAny
NormalRNN	(NotAny; 10; 32).	3104	Sigmoid
Drop out	0.2	0	NotAny
NormalRNN	(NotAny; 10; 16).	784	Tanh
Drop out	0.2	0	NotAny
Densed	(NotAny; 10; 8).	136	NotAny
Densed	(NotAny; 10; 4).	36	NotAny
Densed	(NotAny; 10; 2).	10	NotAny
Densed	(NotAny; 10; 1).	3	Sigmoid

Table 1: Model 1's layers

This model has finished two epochs with a 64-batch batch size. The model's training period can be accelerated by a larger batch size or prolonged by a smaller batch size. The model's loss drops to just below 0.40 after two iterations across the dataset, however its accuracy in transaction identification is lacking. Given that there are equal numbers of legitimate and fraudulent transactions in the dataset of 5,524,392 transactions (using SMOTE)., it has been able to correctly classify half of the fraudulent transactions as fraudulent and



incorrectly identify the remaining half as real. Barely has the loss decreased to 40% (Ata & Hazim, 2020).

Figure 9: Model 1's loss over time

```
TP =  519657                    FP =  32712
FN =  276340                    TN =  276170
% of fraud detected:     0.5 ( 0.49984615663064924 )
% of fraud missed:       0.5 ( 0.5001538433693508 )
Average Precision-Recall score:  0.6970190246996255
```

Figure 10: RNN model 1 (SMOTE). fraud detection and missing

B. Model 2

Model 1 is an excellent place to start, but with too much data, the model can overlook crucial information. This approach addresses the latter issue by utilizing LSTMs. Now, when information moves from one layer to the next, it will be taken into account if it adds to the ultimate judgment.

Layer	Output Shape	Parameters	Activation Function
lstm	(NotAny; 10; 128).	16640	Sigmoid
Drop out	0.2	0	NotAny
lstm	(NotAny; 10; 64).	12352	Sigmoid
Drop out	0.2	0	NotAny
lstm	(NotAny; 10; 32).	3104	Sigmoid
Drop out	0.2	0	NotAny
lstm	(NotAny; 10; 16).	784	Tanh
Drop out	0.2	0	NotAny
Densed	(NotAny; 10; 8).	136	NotAny
Densed	(NotAny; 10; 4).	36	NotAny
Densed	(NotAny; 10; 2).	10	NotAny
Densed	(NotAny; 10; 1).	3	Sigmoid

Table 2: Model 2's layers

Similar to the preceding model, this one has used a batch of 64 data to complete two epochs. However, because the LSTM layers do extra calculations, model 2 took longer to train. Even with a reduced loss of 34%, the model's performance is somewhat poorer than the prior model's. Despite having the same average precision-recall score of 0.69, Model 2 has incorrectly identified almost half of the fraudulent transactions as legitimate. The model maintains its similar performance even with LSTMs.

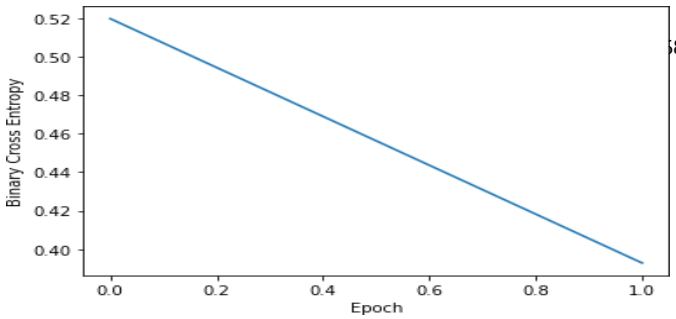


Figure 11: Model 2's loss over time

```
TP =  519657                    FP =  32712

FN =  276340                    TN =  276170

% of fraud detected:     0.5 ( 0.49984615663064924 )

% of fraud missed:       0.5 ( 0.5001538433693508 )

Average Precision-Recall score:  0.6970190246996255
```

Figure 12: RNN model 2 (SMOTE). fraud detection and missing (Hazım, 2018)

C. Model 3

Since Models 1 and 2 have only covered two epochs, it's possible that they will still be unable to distinguish between a legitimate transaction and a fraudulent one. With the exception of having gone through 20 epochs, this model is identical to model 1.

Layer	**Output Shape**	**Parameters**	**Activation Function**
NormalRNN	(NotAny; 10; 128).	16640	Sigmoid
Drop out	0.2	0	NotAny
NormalRNN	(NotAny; 10; 64).	12352	Sigmoid
Drop out	0.2	0	NotAny
NormalRNN	(NotAny; 10; 32).	3104	Sigmoid
Drop out	0.2	0	NotAny
NormalRN	(NotAny;	784	Tanh

N	10; 16).		
Drop out	0.2	0	NotAny
Densed	(NotAny; 10; 8).	136	NotAny
Densed	(NotAny; 10; 4).	36	NotAny
Densed	(NotAny; 10; 2).	10	NotAny
Densed	(NotAny; 10; 1).	3	Sigmoid

Table 3: Model 3's layers

However, the conclusion is not significantly affected by even more epochs. Even if the loss drops by less than 15%, the outcome is the same as the later models.

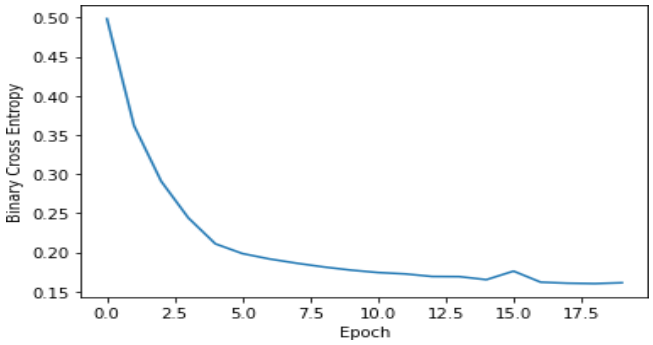


Figure 13: Model 3's loss over time

```
TP =  520017                    FP =  32352

FN =  280137                    TN =  272373

% of fraud detected:     0.49 ( 0.4929738828256502 )

% of fraud missed:       0.51 ( 0.5070261171743498 )

Average Precision-Recall score:  0.6941813133469317
```

Figure 14: RNN model 3 (SMOTE). fraud detection and missing (Hazım, 2018)

D. Model 4

Model 4 runs through 10 epochs while maintaining the same RNN architecture as Model 2.

Layer	Output Shape	Parameters	Activation Function
lstm	(NotAny; 10; 128).	16640	Sigmoid
Drop out	0.2	0	NotAny
lstm	(NotAny; 10; 64).	12352	Sigmoid
Drop out	0.2	0	NotAny
lstm	(NotAny; 10; 32).	3104	Sigmoid
Drop out	0.2	0	NotAny
lstm	(NotAny; 10; 16).	784	Tanh
Drop out	0.2	0	NotAny
Densed	(NotAny; 10; 8).	136	NotAny
Densed	(NotAny; 10; 4).	36	NotAny
Densed	(NotAny; 10; 2).	10	NotAny
Densed	(NotAny; 10; 1).	3	Sigmoid

Table 4: Model 4's layers

Even with extra epochs included, the LSTM model was still unable to correctly identify fraudulent

January, 2024
Volume: 9, No: 1, pp. 1897-1914
ISSN: 2059-6588(Print) | ISSN 2059-6596(Online)

transactions in situations when the loss was less than 20%.

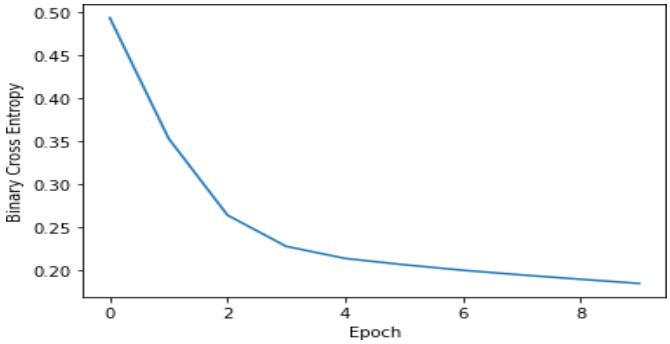


Figure 15: Model 4's loss over time

```
TP =  520000                    FP =  32369

FN =  288752                    TN =  263758

% of fraud detected:     0.48 ( 0.4773814048614505 )

% of fraud missed:       0.52 ( 0.5226185951385496 )

Average Precision-Recall score:  0.6865425238102194
```

Figure 16: RNN model 4 (SMOTE). fraud detection and missing

CONCLUSION

For Green Finance to avoid laundering clients' money, fraud detection is essential. Online transactions have increased in popularity among us throughout the epidemic, and there is now a greater chance of fraud or identity theft. Using the PaySim dataset, this research compares and creates accurate machine learning and deep learning models that may be used to identify fraud. Furthermore, several measures for assessing the performance of an algorithm are addressed Overall, single-choice algorithms (Gaussian Naïve Bayes, Logistic Regression, and Recurrent Neural Networks). do not perform as well as ensemble approaches like Random Forest, XG Boost, and K-Nearest Neighbors. RNNs are the quickest in real-time fraud detection, but additional testing is needed to create an accurate model that can beat the other methods. SMOTE and class weights illustrate the differences and interactions between the algorithms and the data. K Nearest Neighbors illustrates the most significant distinction between the two approaches.

To enhance my writing, I may design a unique loss function for both Traditional Machine Learning and Deep Learning. This function will determine the money lost on False Negative transactions, which occur when a model incorrectly classifies a transaction. Pay Sim is a mobile money simulator with actual administrative charges; thus, this may be difficult. It does not provide any other details you may have about

the client. When determining a Green Finance's entire loss from a fraudulent transaction, administrative expenses like utilities, insurance, payroll, office space, power, etc. are vital. The whole loss in the event that an algorithm incorrectly classifies a transaction is equal to the money lost plus the administrative expenses. True negative outcomes, on the other hand, merely incur administrative costs. Secondly, in order to determine whether or not the RNN models will get better, they might be evaluated using larger batches and more epochs. If there are more epochs or lower batch sizes, the calculation time will rise.

Another task is to analyse and improve the Deep Learning models that are currently being proposed. To create a new RNN model, a great deal of trial and error with various epochs and batch sizes must be done throughout the lengthy training process. Moreover, Due to its sequential data learning nature, RNN may not be the best model for this dataset. By analyzing data like how long it takes a customer to navigate between pages or even how long they spend entering their card information or browsing the website, the RNN can be used to effectively identify patterns in user behavior that could potentially prevent fraudulent transactions from happening. Lastly, more than five neighbors can be used to test the K-Nearest Neighbors method. Slower algorithms are those that evaluate a larger number of neighbors.

REFERENCES

Al-Hashedi, K. G., & Magalingam, P. (2021). Financial fraud detection applying data mining techniques: A comprehensive review from 2009 to 2019. *Computer Science Review*, *40*, 100402.

Alarfaj, F. K., Malik, I., Khan, H. U., Almusallam, N., Ramzan, M., & Ahmed, M. (2022). Credit card fraud detection using state-of-the-art machine learning and deep learning algorithms. *IEEE Access*, *10*, 39700-39715.

Ata, O., & Hazim, L. (2020). Comparative analysis of different distributions dataset by using data mining techniques on credit card fraud detection. *Tehnički vjesnik*, *27*(2), 618-626.

Awoyemi, J. O., Adetunmbi, A. O., & Oluwadare, S. A. (2017). *Credit card fraud detection using machine learning techniques: A comparative analysis.* Paper presented at the 2017 international conference on computing networking and informatics (ICCNI).

Bandyopadhyay, S., Thakkar, V., Mukherjee, U., & Dutta, S. (2021). Emerging approach for detection of financial frauds using machine learning.

Chaudhary, K., Yadav, J., & Mallick, B. (2012). A review of fraud detection techniques: Credit card. *International Journal of Computer Applications*, *45*(1), 39-44.

Cui, J., Yan, C., & Wang, C. (2021). *Learning transaction cohesiveness for online payment fraud detection.* Paper presented at the The 2nd International Conference on Computing and Data Science.

Dileep, M., Navaneeth, A., & Abhishek, M. (2021). *A novel approach for credit card fraud detection using decision tree and random forest algorithms.* Paper presented at the 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV).

Gupta, P., Varshney, A., Khan, M. R., Ahmed, R., Shuaib, M., & Alam, S. (2023). Unbalanced Credit Card Fraud Detection Data: A Machine Learning-Oriented Comparative Study of Balancing Techniques. *Procedia Computer Science*, *218*, 2575-2584.

Hazım, L. R. (2018). *Four classification methods Naïve Bayesian, support vector machine, K-nearest neighbors and random forest are tested for credit card fraud detection.* Altınbaş Üniversitesi,

Höppner, S., Baesens, B., Verbeke, W., & Verdonck, T. (2022). Instance-dependent cost-sensitive learning for detecting transfer fraud. *European Journal of Operational Research*, *297*(1), 291-300.

Karthika, J., & Senthilselvi, A. (2023). Smart credit card fraud detection system based on dilated convolutional neural network with sampling technique. *Multimedia Tools and Applications*, 1-18.

Kaur, H., Pannu, H. S., & Malhi, A. K. (2019). A systematic review on imbalanced data challenges in machine learning: Applications and solutions. *ACM Computing Surveys (CSUR)*, *52*(4), 1-36.

Lakshmi, S., & Kavilla, S. (2018). Machine learning for credit card fraud detection system. *International Journal of Applied Engineering Research*, *13*(24), 16819-16824.

Olivas, E. S., Guerrero, J. D. M., Martinez-Sober, M., Magdalena-Benedito, J. R., & Serrano, L. (2009). *Handbook of research on machine learning applications and trends: Algorithms, methods, and techniques: Algorithms, methods, and techniques*: IGI global.

Pumsirirat, A., & Liu, Y. (2018). Credit card fraud detection using deep learning based on auto-encoder and restricted boltzmann machine. *International Journal of advanced computer science and applications*, *9*(1).

Torres Berru, Y., López Batista, V. F., Torres-Carrión, P., & Jimenez, M. G. (2020). *Artificial intelligence techniques to detect and prevent corruption in procurement: a systematic literature review*. Paper presented at the Applied Technologies: First International Conference, ICAT 2019, Quito, Ecuador, December 3–5, 2019, Proceedings, Part II 1.