Machine Learning Model for Detecting Money Laundering in Bitcoin Blockchain Transactions

Jacob Kehinde Ogunleye
Department of Computer Science Faculty of Science
Adekunle Ajasin University Akungba-Akoko
Ondo State, Nigeria

Olusola Olajide Ajayi
Department of Computer Science Faculty of Science
Adekunle Ajasin University Akungba-Akoko
Ondo State, Nigeria

Adewuyi Adetayo Adegbite
Department of Computer Science Faculty of Science
Adekunle Ajasin University Akungba-Akoko
Ondo State, Nigeria

Joy Rotimi Obafemi
Department of Computer Science Faculty of Science
Adekunle Ajasin University Akungba-Akoko
Ondo State, Nigeria

Olatunde David Akinrolabu
Department of Computer Science Faculty of Science
Adekunle Ajasin University Akungba-Akoko
Ondo State, Nigeria

Akinola Elijah Ebitigha
Department of Computer Science Faculty of Science
Adekunle Ajasin University Akungba-Akoko
Ondo State, Nigeria

ABSTRACT
The problem of money laundering has significantly impacted Nigeria's economy, with the rise of cryptocurrency exacerbating financial crimes like terrorism financing. To address this, a project aimed to develop a machine learning model to detect money laundering in bitcoin transactions using blockchain security technology. The dataset, consisting of 2,906 entries from Kaggle, was split into 70% for training and 30% for testing. Cross-validation trials were also conducted. The k-means clustering algorithm, an unsupervised learning technique, was used to group the data, and the K-Nearest Neighbour (KNN) classifier labeled these clusters. The model predicted bitcoin laundering based on these labeled samples, leveraging blockchain's data immutability through consensus mechanisms and cryptographic principles. Experimental results showed that the algorithm accurately identified 87.2% of legitimate transactions and 12.6% of money laundering operations, with a very low misclassification rate of 0.002%. The model achieved 95% accuracy, 97% precision, and 100% recall using the percentage split approach. Additionally, a 5-fold cross-validation yielded a mean accuracy score of 99%, indicating the model's robustness and reliability without overfitting or underfitting. In summary, the model demonstrated high reliability, accurately distinguishing between legitimate and illicit bitcoin transactions with minimal error.

Keywords
Money laundering, bitcoin transactions, financial crimes, blockchain, crypto currency

1. INTRODUCTION
Money laundering is the process of hiding the source, kind, existence, destination, and use of assets or cash acquired through criminal activity, including drug trafficking, major crimes, embezzlement, and corruption. Through this method, money obtained illegally is converted into assets that are difficult to link to their illegal source through legal trade. Money laundering has gained widespread recognition as a global issue in recent years (Nwala Paul, 2023). It can also mean hiding the names of individuals who collected the cash while using money obtained illegally to produce assets that seem to originate from legitimate sources (Lam, 2022).

Usually, the money laundering procedure consists of three steps:

1. Placement: Illicit funds are deposited into banks using cash or other financial instruments, or they are broken up into smaller, less suspicious amounts, which is how they are first brought into the financial system.
2. Layering: The goal of this stage is to separate the money from their illicit source by entangling them in a web of intricate financial transactions that make it impossible to track down the money. This could entail purchasing and selling assets, transferring money between accounts, transacting internationally, or engaging in other confusing activities.
3. Integration: The money that has been laundered is put back into the market and looks honest and pure. This may entail making investments in legitimate companies, buying properties, or taking part in other actions that cause the money to seem like legitimate revenue (Praveen, 2015).

Money laundering can happen in a number of ways, such as through shell companies, banks, financial institutions, transactions involving real estate, and virtual currencies. As authorities advance their detection techniques, the methods—which are frequently complex—continue to change (Milind et al., 2023).

Governments are required to accept Bitcoin as legal cash for the payment of any debt, including taxes, fees for public or private use, and corporate obligations (Fernando et al., 2022). However, depending on how each nation interprets legal tender, something that is designated as such does not always mean it may be traded for regional goods and services like coins and banknotes. Any money issued by the Federal Reserve is considered legal tender in the United States, although merchants are not compelled to take cash payments. Coins and banknotes are also accepted as legal money in Australia, but business owners are not required to accept them for exchange (Kovanen, 2019).

With the adoption of Bitcoin as legal tender, some Countries see it as a chance to create jobs, offer incentives for investment, and improve financial inclusion. For example, huge Bitcoin holders who invest three BTC in El Salvador, one of the poorest economies in Latin America with 6.5 million people and a GDP of $27 billion, can get permanent residency and avoid capital gains taxes. However, because of its possible drawbacks, Bitcoin has not been accepted as legal tender in many underdeveloped nations where security risks are common. The usage of Bitcoin in any form has been fiercely resisted by nations like Burundi, Algeria, Bangladesh, China, and Iraq (Cryptocurrency Regulation, 2022).

Scammers have entered the payment industry as Bitcoin becomes more and more popular, fraudulently transferring funds to other accounts. Thus, money laundering continues to be a widespread problem that impacts the foundation of national economies all over the world (Mohd, 2016). Financial institutions have faced a major challenge in the form of money laundering, with an estimated $800 billion to $2 trillion laundered annually through the global financial industry, unintentionally funding drug cartels, terrorist networks, and other criminal organizations that pose a threat to national security (Kovanen, 2019).

2. REVIEW OF LITERATURE

"Money Laundering: Concept, Significance, and its Impact" by Vandana (2012). According to Vandana, money laundering is a tactic used to justify large sums of cash earned unlawfully through serious crimes, terrorism, and drug trafficking, among other activities. Through the use of reputable companies, this procedure covers up the money's illicit source. International cooperation is necessary to combat money laundering successfully. International law enforcement authorities must work together to successfully investigate and prosecute these complex criminal networks due to their global reach. According to Vandana, money laundering threatens both political and economic stability, hence it must be totally outlawed. Therefore, in order to destroy these networks, international governments must band together and enact strict legal restrictions.

Andreas and colleagues (2017): An All-inclusive Reference Model for Distributed Ledger Technology Based on Blockchain. A blockchain, according to the authors, is a distributed, transactional database that is accessible to every network node. They emphasize on business-level characteristics including players, roles, services, processes, and data models in their examination of four blockchain platforms. Their research is condensed into a reference model that software engineers, system analysts, and business analysts can use as a guide when creating new blockchain platforms or putting existing ones into use. Users can better grasp the technology and how it works with the aid of this model. The authors discovered that, in supporting attempts to standardize blockchain technology, their reference model covers well-known blockchain technologies including Cryptonote, NXT, Hyperledger, and Tendermint.

Jan and colleagues (2018) on Ensuring Security and Secrecy in Decentralized Bitcoin. In order to investigate a private substitute for the centralized banking system, the researchers looked at Bitcoin, a decentralized digital currency whose popularity has increased. They did point out, though, that new research indicates that individuals can still be recognized, and that payments can be tracked back to the blockchain—Bitcoin's open transaction database. Their research showed that the protocols they created are resistant to attacks by hostile parties and have wider uses, like protecting Bitcoin wallets. The researchers came to the conclusion that while though their work was primarily focused on Bitcoin, it is compatible with other cryptocurrencies that employ the same ECDSA primitive, such as Mastercoin and Litecoin.

Campbell-Verduyn (2018), Global Anti-Money Laundering Governance, Bitcoin, and Crypto-Coins. Campbell-Verduyn talks about how the public and policymakers have been interested in peer-to-peer networks that trade digitally encrypted tokens like Bitcoin. He assesses the effectiveness of global anti-money laundering (AML) initiatives in light of cryptocurrencies and their future possibilities. The author asserts two main points: (i) Because of their usage in illegal operations and the possible advantages of blockchain technology, cryptocurrencies have an impact on global AML initiatives. (

ii) To balance the potential and challenges posed by cryptocurrencies, the Financial Action Task Force (FATF) uses a risk-based methodology. AML initiatives involving cryptocurrencies require constant oversight and ethical considerations, according to Campbell-Verduyn. He comes to the conclusion that the alleged hazards of cryptocurrency to AML initiatives are largely unfounded, corroborating**.**

Wil van der et al. (2021): Bitcoin Abnormal Transaction Detection Based on Machine Learning The authors set out to create a dependable method for identifying unusual bitcoin transactions, which could be utilized for illicit trading and money laundering. They utilized the Elliptic dataset, which contains 46,564 Bitcoin transactions (42,019 legal and 4,545 illicit), to train a machine learning-based algorithm to detect anomalous bitcoin transactions. A variety of hyperparameters and machine learning algorithms were included in their suggested approach. Metrics like accuracy, precision, recall, F1 score, and balanced accuracy were used to assess the model's performance. The productivity score of 0.9780 was attained, however the authors pointed out that this was not enough to reliably identify unusual transactions. They increased the classification accuracy by using the TomekLinks resampling methodology in unbalanced situations; the XGBClassifier method achieved an accuracy of 0.9921.

3. MATERIALS AND METHODS

Employing secondary data allowed the research goals to be met. The primary characteristics (factors) required for this investigation were obtained from Kaggle.com, an online data repository, where the implementation data was obtained. The collection includes assessments of 2,906 samples, with each sample being examined using 24 different qualities.

A. System Architecture

To accomplish its goals, the study adopts an unsupervised machine learning approach called k-means clustering. Figure 3.1 provides an illustration of the system architecture.

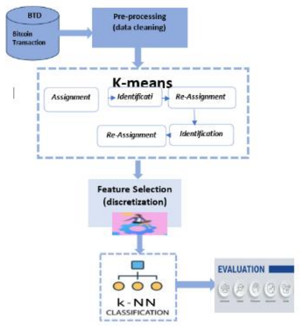


Figure 1: System Architecture

The system architecture consists of the following components:

i. Bitcoin Data

ii. Preprocessing

iii. Clustering Algorithm

iv. Feature Selection

v. Classification

B. Data Preprocessing (Bitcoin Preprocessing)

Cleaning and preparing the data for classification is known as data pre-processing. Internet data frequently has a high level of noise and unhelpful components like scripts, tags, and adverts. Furthermore, a large number of the dataset's samples have little to no effect on its general direction. Keeping these samples adds dimension to the problem and raises the difficulty of classification because each sample is evaluated according to its attributes. As a result, the main goal is to purge the data of all except the most crucial characteristics. The following are the steps in data processing:

a) Data Cleaning

This process entails eliminating superfluous content, such pointless conversations, which may impede the precision of the intended outcomes. The meaningless, unintelligible material that is not machine-readable is removed.

b) Data Transformation

This stage guarantees that the data is in the right format for the categorization procedure.

c) Normalization

This method is used to prepare the data for categorization. The goal is to bring the dataset's numeric column values to a single scale without erasing any information or distorting the value range differences. In mathematics, it is expressed as: $\frac{(X-Xmin)}{Xmax-Xmin}$

i. Scaling – This method will be used to normalize the range of independent variables or features of the data. The goal is to remove skewness. It is mathematically represented as:

$$\frac{(X)}{(n)}$$

$$where\ (n)\ is\ any\ number\ and\ (x)\ is\ the\ numpy$$

$$array\ of\ each\ image$$

C. Mathematical Model

It is reasonable to divide up the data into logical groupings for analysis when working with big data sets. K centroids are first estimated using the k-means technique. These centroids are chosen at random from the dataset. Two phases are alternated iteratively by the algorithm: centroids are updated and data points are assigned.

Outline of the algorithm:

Assuming we have K (the required number of clusters) and input data points X1, X2, X3,... n. We adhere to the protocol listed below:

1. Using a random selection process or by selecting the first K points, choose K starting centroids from the dataset.
2. Determine the Euclidean distance between every data point and the K centroids that have been identified.
3. Using the computed distances, place each data point in relation to the closest centroid.
4. By taking the average of the points in each cluster, find the new centroids.
5. Continue repeating steps 2 through 4 until the centroids stop changing, or for a set number of iterations.

In mathematics, the Euclidean distance between two locations in space is expressed as follows:

$$d(p,q) = \sqrt{(q1-p1)2} + (q2-p2)2$$

Operation of the k-means Algorithm:

1. Find out how many clusters there are (K).
2. To initialize centroids, shuffle the dataset and choose K data points at random to serve as the centroids without substitution.
3. Continue until the centroids do not move, indicating that the clustering of the data points is stable:
   a. Determine the total squared distance between each centroid and each data point.
   b. Assign the closest centroid to every data point.
   c. By averaging each cluster's data points, update the centroids.

The Expectation-Maximization (E-M) strategy is employed by the k-means algorithm to address clustering issues. Assigning data points to the nearest cluster is the E-step; recalculating each cluster's centroid is the M-step. A mathematical explanation of this procedure is provided below (review is optional).

The objective function is:

$$J = \sum_{i=1}^{m} \sum_{k=1}^{k} w_{ik}\ ||x^i - \mu k||.^2 \qquad (1)$$

Hence, if data point xi is a member of cluster k, wik=1; if not, wik = 0. Furthermore, the centroid of xi's cluster is ρk.
There is a two-part minimization problem that the k-means method tackles. Initially, we treat µk fixed and minimize the objective function J with respect to wik. We then treat wik fixed and minimize J with respect to µk. In technical terms, we update cluster assignments (E-step) and differentiate J w.r.t. wik first. Next, we compute the centroids again after the cluster assignments from the previous step (M-step) and differentiate J with respect to µk. Consequently, E-step is:

$$\frac{\partial J}{\partial w_{ik}} \sum_{i=1}^{m} \sum_{k=1}^{k}.\ \|x^i - \mu k\|\ ^2 \rightarrow w_{ik} = \{\frac{1}{0}\ if\ k$$
$$= argmin, \|x^i - \mu_j\|\ otherwise$$

(2)

In simpler terms, based on the sum squared distance of the data point xi from the cluster centroid, designate it to the nearest cluster.

Additionally, M-step is:

$$\frac{\partial J}{\partial \mu k} = 2\sum_{i=1}^{m} w_{ik} \left(x^i - \mu k\right) = 0 \rightarrow \mu k = \frac{\sum_{i=1}^{m} w_{ik} x^i}{\sum_{i=1}^{m} w_{ik}} \quad (3)$$

D. Classification (K-Nearest Neighbors)

Using the group that the data points closest to a given data point belong to, the k-means neighbors (KNN) algorithm is a data categorization technique that estimates the probability that a given data point will join that group or not. In contrast to artificial neural network classification, k-nearest neighbors classification is a straightforward and easy-to-implement method.

It begin by establishing a few definitions and annotations. We'll use y to represent the target and x to represent a feature. This indicates that we are looking for a relationship between x and y in a dataset that has labels for training measurements (x, y). To positively forecast the same observation, x, given an unknown observation, we need to find a function h: X→Y. output y.

We shall first discuss how the KNN classification method operates. The K-nearest neighbor algorithm in the classification problem basically stated that, given a given value of K, we find the K nearest neighbor of the unseen data point. Based on this neighbor, the algorithm assigns the class to the unseen data point, which is the class with the highest number of data points among all the classes of K neighbors. We shall utilize the Euclidean metric for distance metrics. Equation 1

$$d(x, x') = \sqrt{(x_1 - x_1')^2 + \ldots + (x_n - x_n')^2}$$

Finally, the input x gets assigned to the class with the largest probability. Equation 2

$$P(y = j | X = x) = \frac{1}{K} \sum_{i \in \mathcal{A}} I(y^{(i)} = j)$$

The method is still the same for regression, but we take the target values into account rather than the neighboring classes. We take the mean, average, or apply any appropriate function to the target values of the closest neighbors in order to forecast the target value for an unknown data point.

Languages such as Python and R are commonly used to implement the KNN algorithm. The following is the pseudocode for KNN:

Pseudocode for KNN:

1. Load the data.
2. Choose the value of K.
3. For each data point in the dataset:
   i. Calculate the Euclidean distance to all training data samples.
   ii. Store these distances in an ordered list and sort it.
   iii. Select the top K entries from the sorted list.
   iv. Assign a label to the test point based on the majority class among the selected points.
4. End.

E. Performance Evaluation

Five (5) classification machine learning metrics will be used in this study to validate and assess the model.

Table 1 Performance Evaluation metrics

S/N	METRIC	MEANING
i.	Accuracy Score	Model efficiency score (A)
ii.	Precision	Model positive prediction efficiency (P)
iii.	Recall	Model ability (R)
iv.	F1-Score	Precision/recall harmonic mean (F1)
v.	Confusion Matrix	TP, FP, TN, FN ratio to each other (CF)

F. Use-Case-Diagram

Use-case diagrams give an overview of a system's high-level features and scope. They show how the system's actors, or features, interact with one another. These diagrams show the functions of the system and how the actors interact with it; however, they do not disclose how the system is internally designed. The main characteristics of the model are displayed in the figure below.

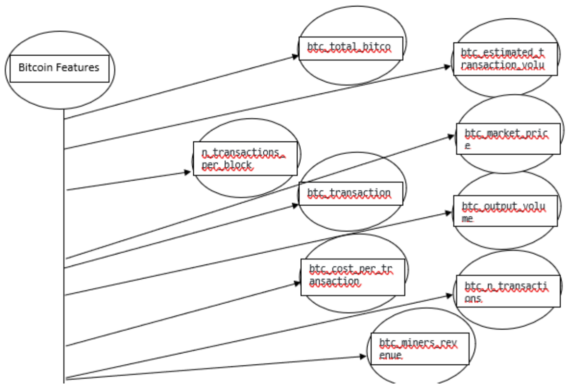


Figure 2: Use-Case Diagram

G. Sequence Diagram

An interaction diagram that shows the steps involved in carrying out an operation is called a sequence diagram. It illustrates how data moves between items in a system. The objects involved in the process are positioned in the message sequence, left to right, based on their respective functions. The suggested system's sequence diagram is shown in Figure 3. Furthermore, Figure 4 presents the flowchart.

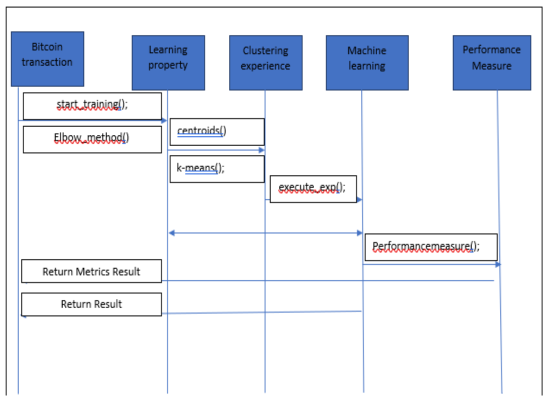


Figure 3: Sequence Diagram

H. System Flow Chart

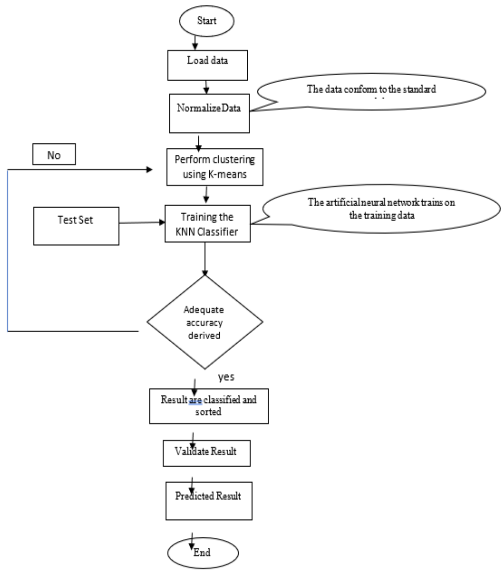
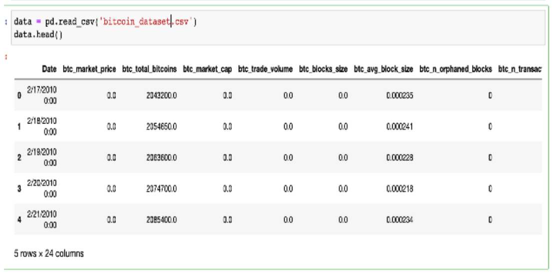


Figure 4 The System Flow chart.

4. RESULTS AND DISCUSSION

In order to create a highly accurate model utilizing a Bitcoin dataset, the implementation portion of this research's practical implementation entails modeling k-means clustering and k-nearest neighbors. Using Jupyter IDE scripts, this is achieved using both supervised (KNN) and unsupervised (k-means) learning techniques. The implementation dataset was further produced using a text editor and preprocessed within the IDE based on predetermined believability criteria.

The process of reading data from a.csv worksheet file and displaying the first five rows of the dataset using a pandas DataFrame in Python is demonstrated in the image below.



```
data = pd.read_csv('bitcoin_dataset.csv')
data.head()
```

	Date	btc_market_price	btc_total_bitcoins	btc_market_cap	btc_trade_volume	btc_blocks_size	btc_avg_block_size	btc_n_orphaned_blocks	btc_n_transac
0	2/17/2010 0:00	0.0	2043200.0	0.0	0.0	0.0	0.000235	0	
1	2/18/2010 0:00	0.0	2054650.0	0.0	0.0	0.0	0.000241	0	
2	2/19/2010 0:00	0.0	2063600.0	0.0	0.0	0.0	0.000228	0	
3	2/20/2010 0:00	0.0	2074700.0	0.0	0.0	0.0	0.000218	0	
4	2/21/2010 0:00	0.0	2085400.0	0.0	0.0	0.0	0.000234	0	

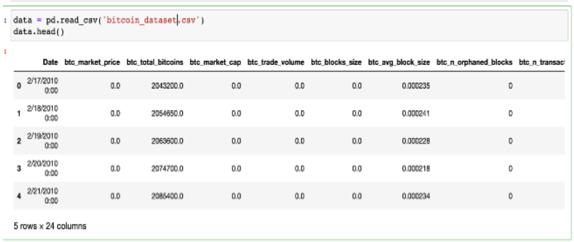
5 rows x 24 columns

Fig 5 Data Downloaded from an online data bank imported into jupyter notebook

Data Training and Testing

When training a machine learning model, backpropagation is used to iterate until the model obtains the lowest possible error. By building a mathematical model, machine learning algorithms are intended to learn from and make predictions about input data. The neural network's weights are adjusted using training data to incorporate both the expected output and the input data, thereby training the algorithm. The training dataset, which is made up of pairs of input vectors and their matching output vectors (or scalars), known as the target or label, is used to fit the model initially. The model is used to train by applying it to the dataset and comparing each input vector's result to the target. Usually, training uses 70% of the data, with the remaining 30% set aside for testing.

The figure below illustrates the process of filling missing values in the dataset.



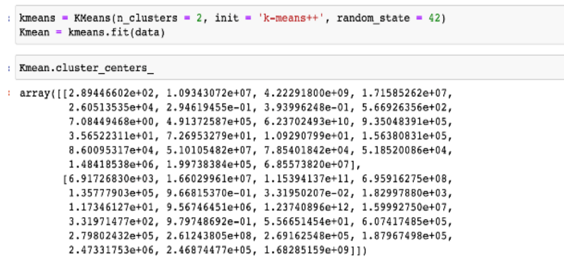
```
data = pd.read_csv('bitcoin_dataset.csv')
data.head()
```

	Date	btc_market_price	btc_total_bitcoins	btc_market_cap	btc_trade_volume	btc_blocks_size	btc_avg_block_size	btc_n_orphaned_blocks	btc_n_transac
0	2/17/2010 0:00	0.0	2043200.0	0.0	0.0	0.0	0.000235	0	
1	2/18/2010 0:00	0.0	2054650.0	0.0	0.0	0.0	0.000241	0	
2	2/19/2010 0:00	0.0	2063600.0	0.0	0.0	0.0	0.000228	0	
3	2/20/2010 0:00	0.0	2074700.0	0.0	0.0	0.0	0.000218	0	
4	2/21/2010 0:00	0.0	2085400.0	0.0	0.0	0.0	0.000234	0	

5 rows x 24 columns

Fig 6: Filling missing values

The figure below depicts the targets obtained from the k-means algorithm (k-clusters), represented as the label class for the dataset..



```
kmeans = KMeans(n_clusters = 2, init = 'k-means++', random_state = 42)
Kmean = kmeans.fit(data)

Kmean.cluster_centers_

array([[2.89446602e+02, 1.09343072e+07, 4.22291800e+09, 1.71585262e+07,
        2.60513535e+04, 2.94619455e-01, 3.93996248e-01, 5.66926356e+02,
        7.08449468e+00, 4.91372587e+05, 6.23702493e+10, 9.35048391e+05,
        3.56522311e+01, 7.26953279e+01, 1.09290799e+01, 1.56380831e+05,
        8.60095317e+04, 5.10105482e+07, 7.85401842e+04, 5.18520086e+04,
        1.48418538e+06, 1.99738384e+05, 6.85573820e+07],
       [6.91726830e+03, 1.66029961e+07, 1.15394137e+11, 6.95916275e+08,
        1.35777903e+05, 9.66815370e-01, 3.31950207e-02, 1.82997880e+03,
        1.17346127e+01, 9.56746451e+06, 1.23740896e+12, 1.59992750e+07,
        3.31971477e+02, 9.79748692e-01, 5.56651454e+01, 6.07417485e+05,
        2.79802432e+05, 2.61243805e+08, 2.69162548e+05, 1.87967498e+05,
        2.47331753e+06, 2.46874477e+05, 1.68285159e+09]])
```

Fig 7: K-means Clusters

The figure below depicts the targets obtained from the k-means algorithm (k-clusters), represented as the label class for the dataset.

```
5 rows x 23 columns

# split data to training set and testing set

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size = 0.3,random_state = 42)
print(y_test.shape)
print(y_train.shape)
print(X_train.shape)
print(X_test.shape)

(872,)
(2034,)
(2034, 23)
(872, 23)
```

Fig 8: Train and Test Dimension

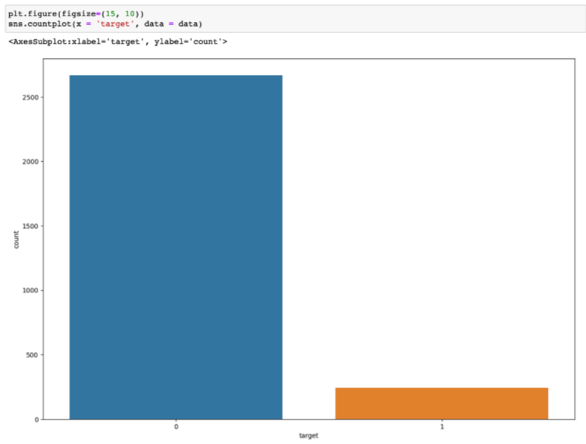


Fig 9: Grouping into their classes

5. RESULT AND ANALYSIS

```
0]: plt.figure(figsize=(20,15))
    sns.heatmap(data.corr(), cmap= "viridis", linecolor='white', annot=True)

0]: <AxesSubplot:>
```

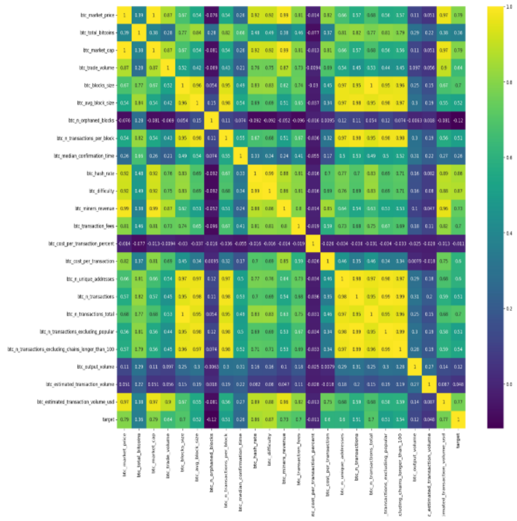


Fig 10: Statistical correlation of the dataset

The statistical correlation of the dataset following its conversion to a numerical dataset is depicted in figure 10 above. Each feature's direct or inverse proportionality to the target class is explained by the correlation, which can be either positive or negative.

Evaluation metrics, including accuracy score, confusion matrix, precision, recall, and f-score, are displayed in figure 11 below. The efficiency and learning capacities of the model are demonstrated by each of these evaluation indicators.

```
model = KNeighborsClassifier(n_neighbors=3)
model.fit(X_train, y_train)
predictionss = model.predict(X_test)
print(" ======== Accuracy Score ======== ")
print(" ")
print(f"{accuracy_score(y_test, predictionss)} or {round(accuracy_score(y_test, predictionss) * 100, 1)}%")
print(" ")
print(" ======== Confustion Matrix ======== ")
print(" ")
print(confusion_matrix(y_test, predictionss))
print(" ")
print(" ======== Classification Report ======== ")
print(" ")
print(classification_report(y_test, predictionss))
```

```
 ======== Accuracy Score ========

0.9977064220183486 or 99.8%

 ======== Confustion Matrix ========

[[802   2]
 [  0  68]]

 ======== Classification Report ========

              precision    recall  f1-score   support

           0       1.00      1.00      1.00       804
           1       0.97      1.00      0.99        68

    accuracy                           1.00       872
   macro avg       0.99      1.00      0.99       872
weighted avg       1.00      1.00      1.00       872
```

Fig 11: Accuracy Score, Confusion matrix and Classification report for the Dataset (Full-set) loaded at once

By adding up all of the misclassified samples (the numbers that are not on the confusion matrix's diagonal) and dividing the result by the total number of test data points, the error created for the full epoch loaded all at once is determined. The following formula is used to determine the overall error the model produces:

$$\text{Error Rate(Full Set)} \quad = \quad \frac{0+2}{872} = 0.0020$$

These evaluation indicators show the effectiveness and degree of learning of the model. The outcomes of data trained and tested with the Naïve Bayes classifier (model) are shown in the above figures. Given that more data enhances learning rate, it is likely that some samples were misclassified because of the small dataset size.

Cross Validation

A model validation approach called cross validation is used to evaluate how well a statistical analysis (model)'s output will generalize to a different set of data. It is mostly used in situations when prediction is the goal and the practical accuracy of the model is being estimated. To make sure the machine learning model is stable and can be applied to new data, validation is crucial. It is critical to confirm that the model maintains low bias and variance by accurately capturing the majority of patterns from the data without overfitting or underfitting.

Cross validation removes the need for a separate validation set by reserving a test set for the final assessment. There are several cross-validation techniques; the most popular is k-fold cross-validation, which is what this study uses. The training set is split into k smaller sets (or folds) for k-fold cross validation.

Cross Validation -->

```
from sklearn.model_selection import cross_val_score, KFold, cross_val_predict

clf1 = KNeighborsClassifier()
model = clf1.fit(X_train, y_train)
predict1 = cross_val_score(model, X, y, cv = 5)
mean = np.mean(predict1)
print(" ")
print("cross validation scores: ", predict1)
print(" ")
print("Mean Score= ", mean)
print(" ")
print(" ")
```

```
cross validation scores:  [0.97766323 1.         1.         1.         0.79862306]

Mean Score=  0.955257258784771
```

Researchers may imitate this process by building trees on a computer using matrices. These networks reduce the complexity by acting as abstractions of individual data points. K-nearest neighbors (KNN) can be customized for particular purposes like pattern recognition or data categorization through a learning process that mainly entails modifying the synaptic connections between data samples.

Table2: Table of Metrics

SN	Model	Accuracy	Precision	Recall	F-Score	Error Rate
1	KNN	99%	97%	100%	100%	0.0020

6. DISCUSSION

The dataset's k-clusters were found using the unsupervised learning model K-means clustering, which also successfully provided class labels for each sample. The refined and labeled data was then used to train and test the KNN classifier using the clustering results. The model predicts and classes bitcoin laundering operations based on the features that were extracted.

At this point, two experiments were carried out. Initially, the k-nearest neighbor classifier's classification accuracy for each data sample was determined. Promising outcomes were obtained when the model was trained and tested utilizing the train-test split technique. With only 2 out of 872 test samples incorrectly categorized, the KNN classifier produced outputs with an error rate of less than 0.0020, yielding 99% accuracy, 97% precision, and 100% recall. Furthermore, a mean accuracy score of 95% was obtained by cross-validation using a 5-fold validation procedure, suggesting that the model is neither overfitted nor underfitted. When it comes to identifying and forecasting bitcoin laundering activity, this model exhibits a high degree of precision and dependability.

7. CONCLUSION

The building of models for precise predictions in a variety of fields, including speech recognition, recommendation systems, and weather forecasting, has been made easier recently by advances in machine learning and deep learning. A machine learning model's accuracy requirements change based on the problem statement, domain, and particular criteria. As such, the process of establishing a model include assessing all pertinent indicators.

The design of the k-means and k-nearest neighbors (KNN) models is the main topic of this paper. Larger datasets and alternative unsupervised learning algorithms, however, can improve these models' performance. The findings show that machine learning model performance is highly influenced by the cleaned datasets chosen for model training.

8. RECOMMENDATION

Secondary data from an online data repository (Kaggle.com) was used in this study to build the model. Future studies can investigate more approachable techniques for money laundering detection using larger and more meticulously cleansed categorical datasets. In addition to using the k-means algorithm for unstructured data labeling and the k-nearest neighbors (KNN) algorithm for supervised learning classification, deeper learning models and other methods could be used to more effectively identify criminal bitcoin transactions.

9. REFRENCES

[1] Nwala Paul (2023), Contextual Issues and Effects of Money Laundering Crimes within the Purview of International Law. *WAUU Journal of International Affairs and Contemporary Studies (WJIACS)* Vol. 3 (2)

[2] Lam Bukeje Ayoker ((2022). Impact of Money Laundering on Economic Growth. *IJRDO - Journal of Business management*. Volume-8 | Issue-1 | Pg1-39.

[3] Praveen Kumar, (2015). Money Laundering in India: Concepts, Effects and Legislation. *International Journal of Research in Humanities & Soc. Sciences [I.F. = 0.352]*. Vol. 3, Issue: 7,July:2015 ISSN:(P) 2347-5404 ISSN:(O)2320 771X. 51-63.

[4] Milind Tiwari, Jamie Ferrill, Adrian Gepp, & Kuldeep Kumar, (2023), Factors influencing the choice of technique to launder funds: The APPT framework, *Journal of Economic Criminology*, Volume 1, Pg. 1-11.

[5] Fernando E. Alvarez, David Argente, and Diana Van Patten, 2022. Are Cryptocurrencies Currencies? Bitcoin as legal Tender in El Salvador.

[6] Kovanen, A. , 2019. Competing With Bitcoin - Some Policy Considerations for Issuing Digitalized Legal Tenders. *International Journal of Financial Research*, 10(4), 1.

[7] Cryptocurrency Regulation. 2022. A History of Financial Technology and Regulation, 129–152. https://doi.org/10.1017/9781316597736.011.

[8] Mohd Yazid bin Zul Kepli Maruf Adeniyi Nasir, 2016. Money Laundering: Analysis On The Placement Methods. *International Journal of Business, Economics and Law*, Vol. 11, Issue 5 (Dec.) ISSN 2289-1552 2.

[9] Kovanen, A. , 2019. Competing With Bitcoin - Some Policy Considerations for Issuing Digitalized Legal Tenders. International Journal of Financial Research, 10(4), 1. https://doi.org/10.5430/ijfr.v10n4p1.

[10] Vandana Ajay Kumar (2012), Money Laundering: Concept, Significance and its Impact. *European Journal of Business and Management,* Vol 4, No.2. Pg. 113-119.

[11] Andreas Ellervee , Raimundas Matulevi˘cius, & Nicolas Mayer (2017), A Comprehensive Reference Model for Blockchain-based Distributed Ledger Technology. *Proceedings of the ER Forum 2017 and the ER 2017 Demo track, Valencia, Spain.*

[12] Jan Henrik Ziegeldorf, Roman Matzutt, Martin Henze, Fred Grossmann & Klaus Wehrle, (2018) Secure and anonymous decentralized Bitcoin mixing, *Future Generation Computer Systems*, Volume 80, 2018, Pages 448-466.

[13] Campbell-Verduyn, M (2018) Bitcoin, crypto-coins, and global anti-money laundering governance. *Crime Law Soc Change* 69, 283–305 https://doi.org/10.1007/s10611-017-9756-5.

[14] Feldman, E. V., Ruchay, A. N., Matveeva, V. K., & Samsonova, V. D. (2021). Bitcoin abnormal transaction detection based on machine learning. *9th International Conference, AIST 2020*, Skolkovo, Moscow, Russia, October 15–16, 2020 Revised Supplementary Proceedings 9 (pp. 205-215). Springer International Publishing.