International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 04 | April - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

Spatio-Temporal Network-based Bank Transactional Behavior Analysis to Detect Suspicious Activities

Dhanush H[1],Priyadharshini S[2],Rithik Kannan V[3]

[1]Student, Department of Information Science and Engineering,

Bannari Amman Institute of Technology, Sathyamnagalm

[2]Student, Department of Information Science and Engineering,

Bannari Amman Institute of Technology, Sathyamangalam

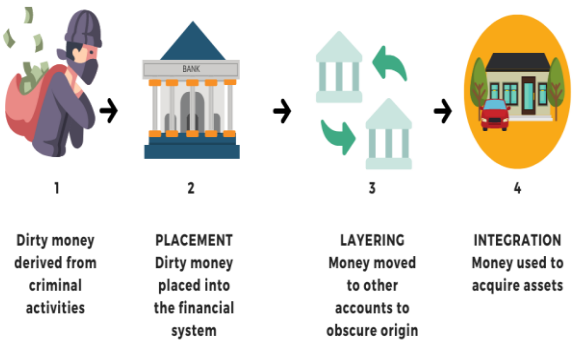
[3]Student, Department of Computer Technology,

Bannari Amman Institute of Technology, Sathyamangalam,

Abstract - Money laundering refers to the act of disguising the proceeds of illegal activities as legitimate funds. This is a significant problem as it enables criminals to profit from illegal activities and finance further criminal endeavors. Money laundering is also linked to other crimes, such as drug trafficking, terrorism financing, and corruption. To combat money laundering, governments and financial institutions have implemented various measures, such as Know Your Customer (KYC) regulations, Anti-Money Laundering (AML) laws, and the use of financial intelligence units. However, the existing money laundering system is complex, making it difficult to detect and prevent money laundering activities. Many current systems rely on outdated technology and manual processes, which can be time-consuming and prone to error. Therefore, there is a need for effective detection and prevention systems that can identify suspicious transactions and patterns of behavior. This project aims to prevent and detect money laundering activities by identifying suspicious transactions and monitoring the movement of funds through the financial system. The proposed system uses Long Short-Term Memory (LSTM) to detect and prevent money laundering activities. By analyzing the data in a time-series format, LSTM can identify unusual patterns of transactions and flag them for further investigation. LSTM can also predict future trends in financial data, allowing for the detection and prevention of potential money laundering activities before they occur.

The proposed transactional network and behavior analysis system provides a more efficient and accurate method for identifying potential money laundering activities. This system can ultimately lead to a more effective and efficient anti-money laundering system.

Keywords: Anti-Money laundering, Long short-term memory, Legitimate funds, Transaction network, behavior analysis, Time series format



1. INTRODUCTION

1.1 Overview

Money laundering is an illegal method of concealing large sums of money that were acquired through criminal activities such as drug trafficking or financing of terrorism. The purpose of this process is to make the money seem like it came from a lawful source, as the money obtained through illegal means is considered "dirty". Money laundering makes it appear as if the money is clean and legitimate.

How Money Laundering Works

Money laundering typically occurs in three phases:

- **Initial entry or placement** is the initial movement of money earned from criminal activity into some legitimate financial network or institution.
- **Layering** is the continuing transfer of money through multiple transactions, forms, investments, or enterprises, to make it virtually impossible to trace the money back to its illegal origin.
- **Final integration** is when the money is freely used legally without the necessity to conceal it any further.

1.2. Problem statement

Many anti-money laundering systems work independently, which makes it difficult to share information and collaborate effectively across different entities and jurisdictions. This hinders the ability to detect and prevent money laundering activities that may involve multiple parties. Traditional anti-money laundering systems often rely on manual processes for transaction monitoring and compliance checks. This can lead to high operational costs, inefficiencies, and delays in detecting suspicious activities. Many existing systems generate a large number of false positive alerts, overwhelming compliance teams and leading to alert fatigue. This makes it challenging to prioritize and investigate genuine suspicious activities effectively.

Some anti-money laundering systems are slow in adopting advanced technologies such as machine learning, artificial intelligence, and natural language processing. These technologies can enhance the detection capabilities and improve the accuracy of identifying potential money laundering activities. Money laundering techniques are constantly evolving, and criminals are finding new ways to exploit vulnerabilities. Existing systems may struggle to keep pace with emerging risks and may not effectively detect new patterns or trends in money laundering activities.

The existing anti-money laundering systems face challenges in effectively detecting and preventing money laundering activities. These challenges include limited integration and collaboration, data quality and timeliness issues, reliance on manual processes, high false positive rates, limited use of advanced technologies, regulatory complexity, emerging risks, and limited international cooperation. These issues result in inefficiencies, increased compliance costs, and an inability to identify and mitigate money laundering risks effectively.

To address these challenges, there is a need for an advanced and intelligent anti-money laundering system. The goal of the "MLBot" system is to leverage artificial intelligence, specifically Long Short-Term Memory (LSTM) networks, to analyze transactional networks and behaviors to detect and prevent money laundering activities. By applying advanced machine learning techniques and behavioral analysis, MLBot aims to improve the accuracy and efficiency of money laundering detection, reduce false positives, enhance data integration and quality, and facilitate effective collaboration among financial institutions and regulatory authorities.

The problem statement of MLBot focuses on developing a comprehensive and intelligent system that can proactively identify suspicious transactions, networks, and patterns indicative of money laundering activities. By leveraging LSTM networks and other advanced technologies, MLBot aims to enhance the effectiveness and efficiency of anti-money laundering efforts, contributing to a more robust and proactive approach to combating financial crime.

1.3. Data science

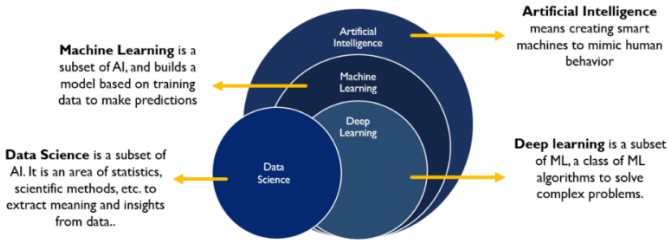
Information Science is the method involved with removing information from organized or unstructured information. It includes different abilities, for example, arithmetic, insights, man-made reasoning, PC programming, perception, and picture investigation. The expression "information science" has been in need for north of thirty years and was at first utilized reciprocally with information examination or information mining. Nonetheless, it has step by step advanced to incorporate more regions past these. The course of information mining includes finding designs in huge informational indexes, similar to when you mine a pile of information and your objective is to track down the chunks of knowledge.

Data Science involves the process of discovering insights from data by examining it at a detailed level to understand complex patterns, trends, behaviors, and inferences. It aims to uncover the necessary information that can assist organizations in making better business decisions. For instance, Netflix analyzes the movie viewing habits of its users to comprehend what motivates their interest and utilizes this information to determine which Netflix series to produce.

Fields of Data Science

Deep learning, machine learning, and artificial intelligence (AI) are just a few of the many fields that fall under the umbrella of data science. Data science analyzes it and draws actionable conclusions from it using a variety of AI, machine learning, and deep learning techniques. Let me explain these terms to you to make things more obvious:

- **Artificial Intelligence:** Artificial Intelligence is a subset of Data Science that enables machines to stimulate human-like behavior.
- **Machine Learning:** Machine learning is a sub-field of Artificial Intelligence that provides machines the ability to learn automatically & improve from experience without being explicitly programmed.
- **Deep Learning:** Deep Learning is a part of Machine learning that uses various computational measures and algorithms inspired by the structure and function of the brain called artificial neural networks.



2. OBJECTIVES AND METHODOLOGY

The objectives of MLBot are as follows:

- To develop an AI-powered transactional network that can identify suspicious patterns of **behaviour** in financial transactions.
- To create a **behaviour** analysis tool that can detect and **analyze** abnormal **behaviour** in financial transactions.
- To provide real-time alerts and notifications to financial institutions and regulatory bodies when suspicious activity is detected.
- To increase the efficiency and accuracy of money laundering detection by automating the process and reducing the risk of human error.
- To reduce the costs associated with money laundering detection and compliance by using AI technology.
- To improve regulatory compliance by providing financial institutions with a tool that can assist them in meeting their legal obligations to detect and prevent money laundering activities.

Data Collection: The process begins by collecting both historical and real-time transactional data, which forms the foundation for subsequent analysis.

Data Preprocessing: Meticulous cleaning involves addressing missing values and outliers. Numerical features are normalized, and categorical variables are encoded for a uniform dataset.

Feature Selection and Extraction: Advanced techniques, including the chi-square test and co-occurrence matrix, are used to extract intricate patterns from data by identifying relevant features.

Model Development using LSTM: The system's core utilizes Long Short-Term Memory (LSTM) for sequence modeling. The model is trained on preprocessed data with fine-tuned hyperparameters to optimize performance. **Deployment:** Post-development, the model integrates seamlessly into the AMLBot system, utilizing an API for continuous monitoring and periodic retraining.

Alert and Notification: The system is designed to provide effective alerts, integrate with regulatory authorities, and offer a user-friendly interface for administrators.

2.1. Deep Learning

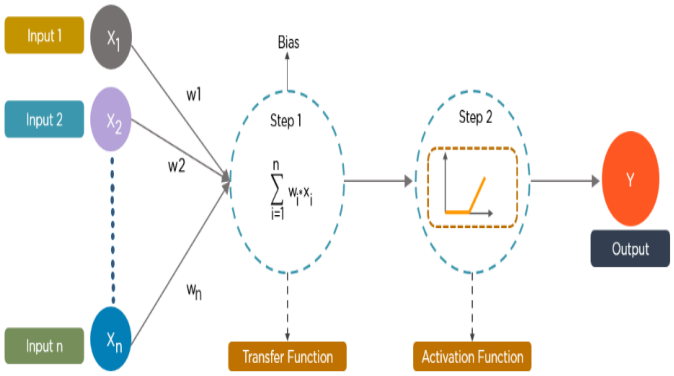
One of the most talked-about areas of machine learning is called "deep learning," which uses sophisticated algorithms based on deep neural networks that are inspired by the structure of the human brain. DL models do not need to be told which features of the input data to prioritize to get accurate results from large volumes of data. Take, for example, the challenge of figuring out which fishing rods on your website get good ratings and which ones get bad ones. Deep neural networks can perform sentiment analysis and extract significant features from the reviews in this kind of situation. A subset of machine learning known as "deep learning" is set apart by the application of highly developed neural networks that were first motivated by the biological neural networks found in human brains. Nodes in linked layers comprise neural networks, which interact with one another to process large amounts of input data.

2.1.1. Importance of Deep Learning

Deep learning algorithms are essential in identifying features and processing vast amounts of structured or unstructured data. However, they may not be suitable for some complex tasks, as they require access to significant amounts of data to function optimally. For instance, a widely used deep learning tool for image recognition, known as Imagenet, employs dataset-driven algorithms that have access to at least 14 million images. This comprehensive tool has set a new benchmark for deep learning tools targeting images as their primary dataset.

There are various types of neural networks such as convolutional neural networks, recursive neural networks, and recurrent neural networks. A typical neural network consists of the input layer, multiple hidden layers, and the output layer that are piled up on top of each other.

A neural network is structured like the human brain and consists of artificial neurons, also known as nodes. These nodes are stacked next to each other in three



layers:

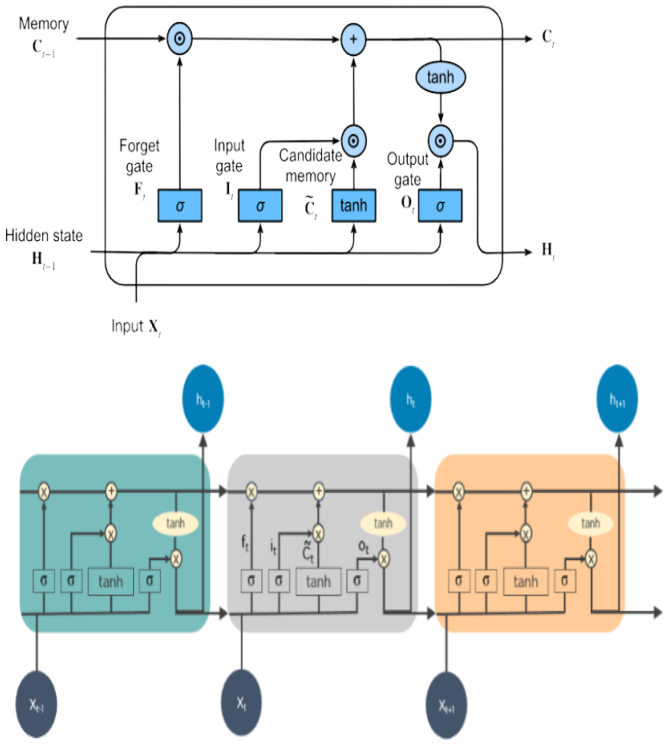
- The input layer
- The hidden layer(s)
- The output layer

Each node receives information from data in the form of inputs. The node computes the inputs, multiplies them by random weights, and adds a bias. Lastly, to choose which neuron to fire, nonlinear functions—also referred to as activation functions—are used.

2.1.2. Long Short-Term Memory Networks (LSTMs)

Long Short-Term Memory (LSTM) can be defined as a type of Recurrent Neural Network (RNN) that is designed to learn and adapt to long-term dependencies in data. LSTMs are capable of memorizing and recalling past data for longer periods compared to other RNNs. They are commonly used in time series predictions as they can retain memory of previous inputs. This is due to their unique four-layered chain-like structure that allows for different types of communication between the layers. Apart from time series prediction, LSTMs have a wide range of applications including speech recognition, pharmaceutical research, and music composition.

LSTM works in a sequence of events. First, they don't tend to remember irrelevant details attained in the previous state. Next, they update certain cell-state values selectively and finally generate certain parts of the cell-state as output. Below is the diagram of their operation.



Over time, LSTMs hold onto information. Their ability to retain past inputs makes them valuable for time-series prediction. Four interacting layers in an LSTM structure are like a chain and communicate uniquely. LSTMs are commonly employed in speech recognition, music creation, and pharmaceutical research, in addition to time-series prediction applications.

3. PROPOSED WORK AND MODULES

MLBot is a proposed system that uses artificial intelligence, specifically Long Short-Term Memory (LSTM) networks, to detect potential money laundering activities by analyzing transactional data. The system aims to capture long-term dependencies and sequential patterns in transactional behaviour, enabling more accurate and timely detection of suspicious activities. MLBot processes sequential transactional data using LSTM, a type of recurrent neural network (RNN) that models temporal dependencies and retains long-term information effectively. To train the LSTM model, the system analyzes transactional features such as transaction amounts, frequencies, timestamps, and relationships between entities. The model learns to identify patterns that are indicative of money laundering, and generates alerts when suspicious activity is detected.

LSTM

LSTM (Long Short-Term Memory) is a type of recurrent neural network (RNN) architecture that is well-suited for modeling sequential data, such as transactional data in the context of anti-money laundering (AML). The LSTM algorithm addresses the vanishing gradient problem often encountered in traditional RNNs by introducing memory cells and gating mechanisms.

Here's a description of the LSTM algorithm used in MLBot:

Memory Cells: A particular kind of neural network called Long Short-Term Memory (LSTM) employs memory cells to retain and transfer data over time. These memory cells enable the model to recognize long-term dependencies in sequential data and assist it in recalling previous inputs. To process and evaluate the incoming data, each memory cell in the LSTM

network has an internal state and communicates with the others. **Gates**: LSTM uses three gates (input, forget, output) with activation functions to control information flow.

Input Gate: The amount of current input that should be saved in the memory cell is decided by the input gate. To generate values between 0 and 1, it employs a sigmoid activation function while accounting for the current input and the previous hidden state.

Forget Gate: An essential part of a memory cell's operation is the forget gate. Determining which data should be removed from the cell is its primary goal. It accomplishes this by combining the input from the current and the hidden state from the past and then applying a sigmoid activation function to this combined data. The amount of information that should be erased from the memory cell is determined by the function's output.

Output Gate: Controlling the amount of data that is released from the memory cell is the job of the output gate. It applies a sigmoid activation function to the input and the hidden state, taking into account both the past and the present. The output is then obtained by passing the result through a tanh activation function.

Hidden State: The output produced at every time step in an LSTM network is the hidden state. It serves as the input for the subsequent step and stores data from the preceding time steps. The output of the memory cells and the output gate are used to calculate the hidden state.

Training: Labeled transactional data, which includes both legal transactions and cases of money laundering, is used to train the LSTM model. To reduce the discrepancy between predicted outputs and true labels, the model modifies the weights and biases of the LSTM layers during training. Gradient descent optimization and backpropagation are the processes involved in this. The LSTM algorithm may successfully capture and preserve important information from previous time steps by employing memory cells and gating mechanisms. Because of its capacity to learn from and recognize trends in sequential transactional data, MLBot can effectively evaluate transactional behavior and detect possible money laundering operations.

- **Dataset**

The LSTM model is trained using a comprehensive dataset of historical transactional data, comprising different transaction attributes and verified cases of money laundering. The dataset comprises both valid transactions and labeled cases of money laundering, which is used for supervised learning. It is important that the dataset is representative of real-world transactional behavior and covers a broad range of money laundering scenarios.

3.1.MLBot Web App

1.1. Front End

The MLBot web app module's front-end development involves designing the user interface and visual elements that users will interact with. Python Flask is utilized as the web framework to build the application's front end. Flask provides a simple and efficient way to develop web applications using Python. To create the user interface, design layouts, and add interactivity to the web app, HTML, CSS, and JavaScript are used. The front-end components are responsible for displaying the MLBot system's features and functionality to the users. The MLBot web app module's front-end development involves designing the user interface and visual elements that users will interact with. Python Flask is utilized as the web framework to build the application's front end. Flask provides a simple and efficient way to develop web applications using Python. To create the user interface, design layouts, and add interactivity to the web app, HTML, CSS, and JavaScript are used. The front-end components are responsible for displaying the MLBot system's features and functionality to the users.

3.1.2. Back End

The back-end development of the MLBot web app module is responsible for processing user requests, executing business logic, and communicating with the database. Python Flask is used for back-end development because it provides a sturdy framework for creating web applications. The back-end components handle the user's inputs, interact with the MLBot system's functionality, and generate responses to be displayed on the front end. They are responsible for coordinating the various modules of the system and ensuring its smooth operation.

3.1.3. Database

The MLBot web app module uses MySQL as its database. MySQL is a widely used relational database management system that offers a secure and scalable solution for storing and fetching data. The database's primary function is to store transactional data, user information, and other pertinent data essential for the MLBot system to function correctly. It ensures efficient data retrieval and manipulation, enabling the system to process and analyze large amounts of data effectively.

3.2. End User Interface

End User Interface Module Description of MLBot Web App:

3.2.1. Admin Interface

The MLBot web app has an admin interface module that provides a user-friendly interface for administrators to efficiently manage and monitor the MLBot system. The module enables admins to perform various tasks such as data collection, importing and exploring datasets, preprocessing data (handling null values, missing values, redundant rows or columns), feature selection using the chi-square test, feature extraction using co-occurrence matrix, building and training the LSTM model for classification, and generating reports.

The admin interface module includes forms and input fields where admins can input dataset files, set pre-processing parameters, select features, and initiate the model training process. It also provides visualizations and data analysis tools to explore and visualize the dataset, helping admins gain insights into the data. Admins can analyze the performance metrics of the LSTM model and adjust the system settings as needed.

3.2.2. MLBot API Interface

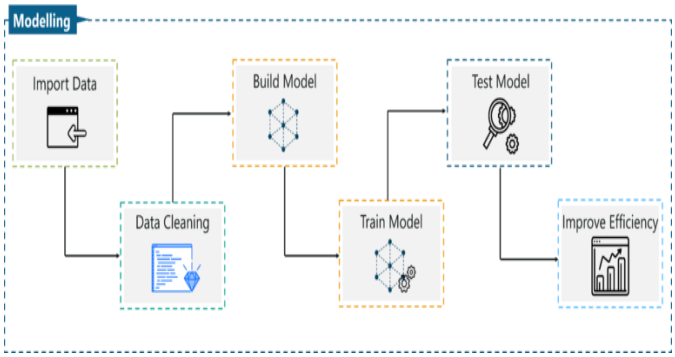
The MLBot API interface module is an essential component of the web app that enables continuous transaction processing and interaction with the MLBot system. It provides external systems or applications with a way to input bank transactions for analysis and prediction. The API interface receives transaction data, validates and pre-processes it, and passes it to the MLBot system's classification model for analysis. Based on the predicted results, it generates alerts and notifications that are sent to the relevant authorities such as RBI and the Income Tax Department.

The MLBot API interface module includes endpoints or routes that external systems can access to send transaction data in a specified format. It streamlines the process of data management, analysis, and alert generation, enabling efficient detection and prevention of money laundering activities. It ensures timely and automated reporting of potential money laundering activities.

Overall, the end-user interface modules of the MLBot web app, which include the admin interface and the MLBot API interface, provide intuitive and convenient interfaces for administrators and external systems to interact with the MLBot system. These interfaces allow for effective and efficient detection and prevention of money laundering activities.

3.3. Money Laundering Activity Classification

The Build and Train module of the MLBot web app is responsible for constructing and training the machine learning model used for classifying money laundering activities. This module utilizes the LSTM (Long Short-Term Memory) algorithm, which is a type of recurrent neural network (RNN) known for its ability to capture long-term dependencies in sequential data.



3.3.1. Data Collection

This dataset contains transactional data that is related to suspected money laundering activities. The data includes various features such as transaction amount, timestamp, transaction type, account numbers, and other relevant attributes. The dataset aims to provide a representative sample of transactions that may exhibit patterns or characteristics associated with money laundering. The dataset is typically provided in a structured format, such as a CSV (Comma-Separated

Values) file, where each row represents a transaction, and each column corresponds to a specific attribute or feature of the transaction. The data has been anonymized to protect the identities and privacy of the individuals and entities involved in the transactions.

Dataset description

1. step - maps a unit of time in the real world. In this case, 1 step is 1 hour. Total steps 744 (30 days' simulation).
2. type - CASH-IN, CASH-OUT, DEBIT, PAYMENT and TRANSFER.
3. amount - the amount of the transaction in local currency.
4. nameOrig - a customer who started the transaction
5. oldbalanceOrg - initial balance before the transaction
6. newbalanceOrig - new balance after the transaction
7. nameless - customer who is the recipient of the transaction
8. oldbalanceDest - initial balance recipient before the transaction. Note that there is no information for customers that start with M (Merchants).
9. newbalanceDest - new balance recipient after the transaction. Note that there is no information for customers that start with M (Merchants).

3.3.2. Import and Explore Dataset

The Import and Explore Dataset module is a crucial component of the MLBot system, responsible for importing the transactional dataset and performing initial exploration and visualization to gain insights into the data. This module sets the foundation for subsequent pre-processing, feature selection, and model-building stages.

	Unnamed: 0	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldb
0	0	7	PAYMENT	11396.93	C512428725	0.00	0.00	M2129874611	0.00
1	1	1	PAYMENT	1461.06	C641030345	226.00	0.00	M521466380	0.00
2	2	6	TRANSFER	18554.73	C248447628	1018.00	0.00	C1955948959	0.00
3	3	702	CASH_OUT	573266.42	C1802792293	573266.42	0.00	C203516640	0.00
4	4	413	CASH_OUT	781.93	C823598101	781.93	0.00	C1338255129	0.00
...	...	...	...	...	...	...	...	...	...
19995	19995	1	PAYMENT	8948.03	C788905599	11137.00	2188.97	M1678709153	0.00
19996	19996	36	CASH_OUT	819503.92	C510717698	819503.92	0.00	C1469248112	255
19997	19997	6	TRANSFER	85354.69	C558400671	85354.69	0.00	C527482085	0.00
19998	19998	383	TRANSFER	106601.15	C1575202335	106601.15	0.00	C278215975	0.00
19999	19999	7	CASH_IN	294780.75	C1795014868	4097511.35	4392292.10	C2040603986	835

The module follows the following steps to import and explore the dataset:

- **Data Import**

The system imports the collected dataset in a structured format like CSV or database table. The import process is handled by the module to ensure accessibility for further analysis.

- **Data Exploration**

The purpose of this module is to perform exploratory data analysis (EDA) techniques to gain a comprehensive understanding of the dataset. This involves examining the structure, features, and statistical characteristics of the dataset. Descriptive statistics such as mean, median, and standard deviation are computed to understand the distribution of numerical variables. Additionally, categorical variables are analyzed to identify unique categories and their frequencies.

- **Data Visualization**

The module makes use of data visualization techniques to visually display the patterns, trends, and relationships present in the dataset. Different

	Unnamed: 0	step	amount	oldbalanceOrg	newbalanceOrig	oldbalanceDest	newbalanceDest
Unnamed: 0	1.000000	-0.006249	-0.009464	-0.004948	0.000168	0.004850	-0.000617
step	-0.006249	1.000000	0.340895	0.146410	-0.100326	-0.036039	0.030994
amount	-0.009464	0.340895	1.000000	0.698877	0.133697	0.003907	0.244926
oldbalanceOrg	-0.004948	0.146410	0.698877	1.000000	0.796172	0.067506	0.179667
newbalanceOrig	0.000168	-0.100326	0.133697	0.796172	1.000000	0.120475	0.076752
oldbalanceDest	0.004850	-0.036039	0.003907	0.067506	0.120475	1.000000	0.896061
newbalanceDest	-0.000617	0.030994	0.244926	0.179667	0.076752	0.896061	1.000000

visualization libraries such as Matplotlib or Seaborn in Python can be utilized to produce plots, histograms, bar charts, scatter plots, and other visualizations. This aids in identifying potential anomalies, clusters, or any other significant patterns that may indicate money laundering activities.

- **Data Summary**

The module generates a report summarizing the key characteristics and observations of the dataset. This report includes information such as the number of records, data types, missing values, and basic statistics of numerical variables. It helps in understanding the quality of the dataset, identifying potential data issues, and guiding further preprocessing steps.

The Import and Explore Dataset module is crucial for gaining initial insights into transactional data, identifying patterns, and understanding the structure and characteristics of the dataset. This information serves as the basis for subsequent preprocessing, feature selection, and model-building steps in the MLBot system.

3.3.3. Pre-processing

The Pre-processing module plays a crucial role in the MLBot system by handling the cleaning and transformation of the transactional dataset before it can be used for further analysis and model building. This module ensures that the data is prepared in a suitable format and quality for accurate and effective money laundering detection.

The Pre-processing module follows the following steps:

- **Handling Null Values**

The module identifies and handles missing values in the dataset. Techniques such as imputation or removal of rows or columns with missing values are applied to ensure data completeness.

- **Handling Missing Values**

The module addresses any missing values present in the dataset. This can be done through techniques such as imputation, where missing values are replaced with estimated or calculated values based on the surrounding data.

- **Handling Misspelled Data**

If the dataset contains misspelled or inconsistent data, the module performs data cleansing techniques to correct the misspelled entries or standardize the data format. This ensures consistency and accuracy in subsequent analysis.

- **Redundant Rows or Columns**

The module identifies and removes any redundant rows or columns that do not contribute meaningful information to the analysis. Reducing redundancy improves computational efficiency and focuses on the relevant features.

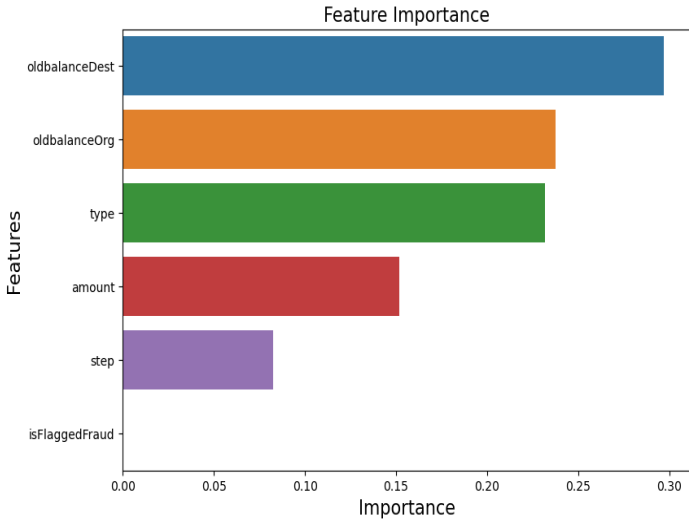
- **Data Transformation**

The Pre-processing module is a crucial step for ensuring the quality of data and preparing the transactional dataset for further analysis. It applies data transformation techniques such as scaling, normalization, or encoding to ensure that the data is in a suitable format for analysis. This step is essential in handling variables with different scales and ensuring that all variables are comparable and meaningful.

The pre-processing module addresses issues such as missing values, misspelled data, and redundant information and transforms the data into a suitable format. By performing these pre-processing steps, the module ensures that the data used for feature selection, feature extraction, and model building is accurate, consistent, and ready for effective money laundering detection using LSTM algorithms.

3.3.4. Feature Selection

The Feature Selection module in the MLBot system employs various formulas and equations to identify the most relevant features from the pre-processed transactional dataset. This module aims to select a subset of features that will optimize the performance of the LSTM model for money laundering detection.



The Feature Selection module follows the following steps, utilizing different formulas and equations:

1. **Chi-square Test**: The module applies the Chi-square test to measure the dependency between each categorical feature and the target variable (legitimate or money laundering). The formula for the Chi-square test statistic is:

$$\chi^2 = \Sigma (O_i - E_i)^2 / E_i$$

Where:

- χ^2 is the Chi-square test statistic.
- O_i is the observed frequency for each category.
- E_i is the expected frequency for each category, assuming no dependency between the feature and the target variable.

The Chi-square test helps determine the significance of the relationship between categorical features and the target variable, indicating their importance in money laundering detection.

2. **Information Gain**: The module calculates the Information Gain for each feature to measure its relevance to the target variable. The formula for Information Gain is:

$$IG(X) = H(T) - H(T|X)$$

Where:

- $IG(X)$ is the Information Gain of feature X.
- $H(T)$ is the entropy of the target variable.
- $H(T|X)$ is the conditional entropy of the target variable given feature X.

Information Gain quantifies the reduction in entropy achieved by considering a particular feature, indicating its discriminatory power in distinguishing money laundering activities.

3. **Recursive Feature Elimination**: The module applies Recursive Feature Elimination (RFE), which recursively eliminates less important features based on their coefficients in a trained model. The RFE algorithm utilizes the following equation:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n$$

Where:

- y is the target variable.
- $\beta_0, \beta_1, \beta_2, ..., \beta_n$ are the coefficients associated with each feature $x_1, x_2, ..., x_n$.

By iteratively eliminating features with the lowest coefficients, RFE identifies the subset of features that contribute the most to the money laundering detection model's accuracy.

The Feature Selection module leverages these formulas and equations to determine the most relevant features for money laundering detection. By utilizing statistical tests, information gain, and recursive elimination techniques, this module ensures that the LSTM model is trained on a subset of features that optimize its performance and enhance the system's ability to detect and prevent money laundering activities.

3.3.5. Feature Extraction

Difference in balance: It is a universal fact that there is never a cent difference between the amount debited from the sender's account and the amount credited to the recipient's account. However, what happens if there is a discrepancy between the amount credited and debited? Some may result from the fees that the service providers charge, but we must draw attention to these odd situations.

Surge indicator: Additionally, whenever a transaction involves a significant sum of money, we must raise a red flag. We deduced that there are numerous outliers with large transaction amounts from the amount distribution. Therefore, we use 450k, the 75th percentile, as our threshold and anything above that will raise a red alert.

Frequency indicator: Here, the user is flagged rather than the transaction. A receiver who receives money

from numerous sources may act as a catalyst for some illicit games of chance or luck. Therefore, when a recipient receives money more than 20 times, it is detected.

Merchant indicator: Since the customer IDs in the receiver begin with "M," it is clear that the users are merchants and that there will be many receiving transactions. Thus, we also indicate whenever a merchant receiver is present.

The Feature Extraction module in the MLBot system utilizes the co-occurrence matrix technique to extract relevant features from the pre-processed transactional dataset. This module aims to capture the relationships and patterns between different elements in the dataset, providing valuable information for money laundering detection.

Weight	Feature
0.6830	step
0.1120	oldbalanceOrg
0.0946	newbalanceOrig
0.0438	type_PAYMENT
0.0160	type_CASH_OUT
0.0156	amount
0.0110	nameDest
0.0110	oldbalanceDest
0.0085	type_TRANSFER
0.0043	newbalanceDest
0.0003	type_CASH_IN
0	type_DEBIT
0	isFlaggedFraud
0	nameOrig

The Feature Extraction module follows the following steps, employing the co-occurrence matrix technique:

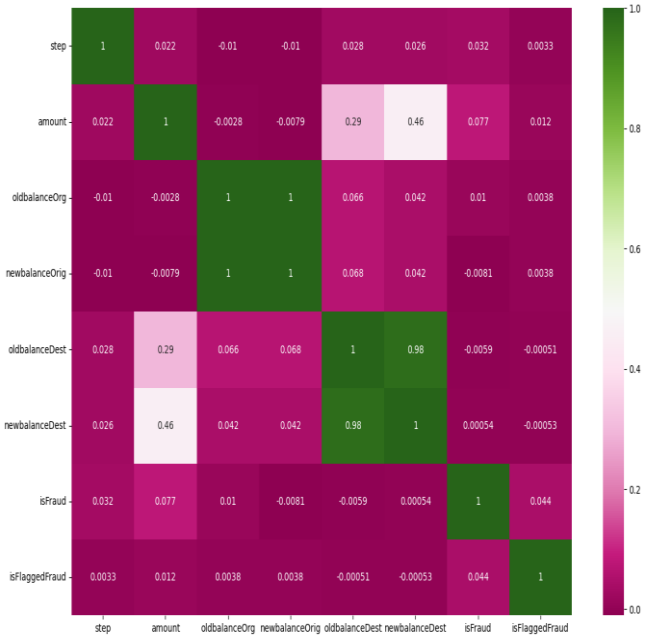
1. **Co-occurrence Matrix:** The module constructs a co-occurrence matrix, which is a square matrix that represents the frequencies of co-occurrence between pairs of elements in the dataset. The formula to calculate the co-occurrence matrix is:

   $M(i, j) = \Sigma C(i, j)$

   Where:

   - M(i, j) is the co-occurrence matrix entry at row i and column j.
   - C(i, j) is the count of co-occurrence between elements i and j.

The co-occurrence matrix captures the frequency of occurrence and the spatial relationships between different elements, revealing important patterns and associations within the dataset.



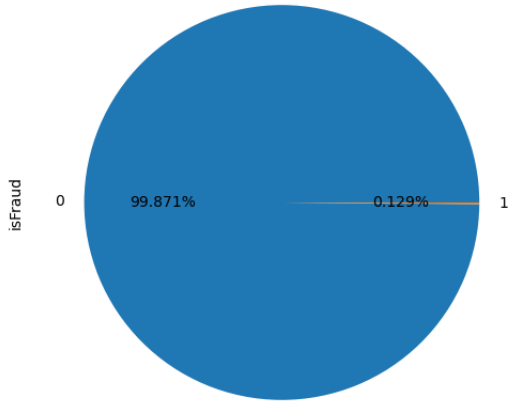
2. **Statistical Measures:** From the co-occurrence matrix, the module extracts various statistical measures to quantify the relationships and patterns. Some commonly used measures include:
   - Correlation: Measures the linear relationship between pairs of elements. It provides insight into how changes in one element correspond to changes in another.
   - Contrast: Measures the local differences between pairs of elements. It indicates how distinct or dissimilar the values of two elements are.
   - Energy: Represents the sum of squared elements in the co-occurrence matrix. It quantifies the overall strength or magnitude of the relationships.

The statistical measures mentioned here help detect different aspects of the relationships between elements, which makes the features more effective in detecting money laundering activities. The Feature Extraction module utilizes the co-occurrence matrix technique and statistical measures to extract meaningful features from the transactional dataset. This technique captures the relationships and patterns between elements, which enriches the input of the LSTM model and improves its ability to detect and prevent money laundering activities with greater efficiency.

3.3.6. Classification

1. The MLBot Web App's classification module uses the Long Short-Term Memory (LSTM) algorithm to identify bank transactions as either legitimate (Class 0) or laundering (Class 1). LSTM is a type of recurrent neural network (RNN) that can capture sequential patterns and dependencies in data. Below is a summary of the classification module and its underlying equations:

2. **LSTM Architecture:**
   - LSTM is composed of LSTM units that process sequential input data.
   - Each LSTM unit has a cell state (Ct) and hidden state (ht) that are updated over time.
3. **Input Representation**:
   - The input to the LSTM consists of pre-processed transactional data, including selected features and transformed data.
   - Each transaction is represented as a sequence of feature vectors.
4. **LSTM Computation**:
   - At each time step t, the LSTM unit receives the input feature vector and computes the output hidden state ht and cell state Ct.
   - The computation involves several mathematical operations, including element-wise multiplications, additions, sigmoid activations, and hyperbolic tangent activations.
5. **Classification Output**:
   - After processing all the input time steps, the final hidden state hT is passed through a fully connected layer with a sigmoid activation function.
   - The sigmoid function outputs a value between 0 and 1, representing the probability of the transaction belonging to Class 1 (Laundering).
   - A threshold is applied to the output probability to determine the final classification label.



The LSTM algorithm in the MLBot Web App learns from the sequential patterns and behaviours in the transactional data to make accurate predictions about money laundering activities. By leveraging the LSTM's ability to capture long-term dependencies, the system can effectively identify suspicious transactions and mitigate the risks associated with money laundering.

3.3.7. Build and Train

The Build and Train module in the MLBot Web App is responsible for constructing and training the LSTM model using the pre-processed and feature-selected dataset. This module follows a series of steps to build the model architecture, compile it with appropriate settings, and train it on the data. Here is a description of the Build and Train module:

Model Architecture:

- The LSTM model is constructed using the Keras library, which provides a high-level interface for building neural networks.
- The model architecture consists of one or more LSTM layers, with additional layers (e.g., fully connected layers) added as needed.
- Experimentation and model performance are used to determine the number of LSTM layers, the number of LSTM units in each layer, and other hyperparameters.

Compilation:

- Before training the model, it needs to be compiled with appropriate settings.
- The compilation step includes specifying the loss function, optimizer, and evaluation metric.

- The binary cross-entropy loss function is frequently utilized for binary classification issues in the MLBot Web App.
- The optimizer, such as Adam or RMSprop, is selected to optimize the model's weights during training.
- The evaluation metric, typically accuracy or AUC-ROC, is used to monitor the model's performance during training.

Training:

- The pre-processed dataset is fed to the LSTM model during training, and the model's weights are iteratively updated.
- The dataset is separated into validation and training sets so that overfitting may be avoided and the model's performance can be tracked.
- The backpropagation algorithm, which computes gradients and modifies weights to minimize the loss function, is used to train the model.
- Several epochs are usually used in the training process, where an epoch is a full run of the dataset. The Build and Train module ensures that the LSTM model is constructed, compiled, and trained properly to achieve accurate and reliable predictions for detecting and preventing money laundering activities.

3.4. Money Laundering Prediction

The MLBot API module in the system serves as the interface for interacting with the trained LSTM model and performing real-time predictions, alerts, and notifications. Here is a description of the MLBot API module:

3.4.1. Input Bank Transactions Continuously

- The MLBot API continuously accepts input bank transactions from various sources, such as financial institutions or transaction monitoring systems.
- These transactions can be in the form of structured data, such as CSV files or API requests, containing information about the transaction, such as amount, date, origin, and destination.

3.4.2. Transaction Pre-processing

- Upon receiving the bank transactions, the MLBot API performs pre-processing steps to prepare the data for the prediction phase.
- This includes handling missing values, handling outliers, standardizing numerical features, and encoding categorical features if required.
- The pre-processing ensures that the input data is in the appropriate format and aligns with the data used during model training.

3.4.3. Prediction

- After pre-processing, the MLBot API feeds the pre-processed bank transactions into the trained LSTM model for prediction.
- The LSTM model analyses the transactional network and behaviour patterns to classify each transaction as either legitimate (Class 0) or indicative of money laundering activities (Class 1).
- The LSTM model uses its learned parameters and the input transaction data to calculate the probability of the transaction belonging to each class.

The MLBot API module acts as a link between real-time bank transactions and the LSTM model that has been trained. Its primary function is to monitor and analyze transactions continuously. It can also send alerts and notifications to regulatory bodies to prevent and investigate any money laundering activities.

3.35. Alert or Notify

The Alert or Notify module is responsible for generating alerts and notifications to the relevant authorities, such as the RBI (Reserve Bank of India) and the Income Tax Department, based on the predictions made by the LSTM model. Here is a description of the Alert or Notify module:

3.5.1. Alert Generation

- When the LSTM model predicts a transaction as indicative of money laundering activities (Class 1), the

Alert module generates an alert to notify the regulatory authorities.

- The alert includes relevant information about the suspicious transaction, such as the transaction details, customer information, and associated risk score.
- The information is extracted from the input bank transactions and the prediction results of the LSTM model.

3.5.2. Notification:

- The Notification module sends the generated alerts to the designated recipients, such as the RBI and the Income Tax Department, to ensure timely action is taken.
- Notifications can be sent through various communication channels, such as email, SMS, or API integrations, based on the preferred method of communication.
- The notifications contain the details of the suspicious transaction, allowing the authorities to review the information and initiate appropriate actions, such as investigations or audits.

3.5.3. Customer Details:

- In addition to the transaction-related information, the Alert or Notify module includes customer details in the alerts and notifications.
- Customer details may include information such as customer ID, name, contact details, account information, and any additional relevant information available.
- Including customer details helps the regulatory authorities identify and track individuals involved in potential money laundering activities.

The Alert or Notify module plays a crucial role in the MLBot system by ensuring that the relevant authorities are promptly informed about suspicious transactions.

4. RESULTS AND DISCUSSION:

The research paper "MLBot: An AI-Powered Transactional Network and Behaviour Analysis to Detect and Prevent Money Laundering Activities using LSTM" includes a "Results and Discussion" section that analyzes, interprets, and evaluates the system's performance after it has been put into place. This section's primary objective is to shed light on how well the MLBot system works to identify and stop money laundering activity.

This typically includes the following components:

4.1.1Results Analysis:

- • Displaying the assessment metrics derived from the performance analysis, such as recall, accuracy, precision, and F1-score.
- Discussion on the performance of the MLBot system in classifying transactions as legitimate or potential money laundering activities.
- Comparison of the results with the desired system goals and objectives.
- Identification of any limitations or challenges faced during the analysis.

4.1.2.Interpretation of Findings:

- Discussion on the significance of the results and their implications for detecting and preventing money laundering activities.
- Identification of patterns, trends, or key features that contribute to the successful detection of money laundering transactions.
- Analysis of false positives and false negatives to identify potential areas of improvement in the system.

4.1.3.Discussion of System Limitations:

- Identification and discussion of any limitations or constraints of the MLBot system.
- Consideration of factors that may impact the accuracy and effectiveness of the system, such as data quality, sample size, or algorithm performance.

- Suggestions for possible enhancements or future research directions to address the identified limitations.

4.1.4.Validation and Reliability:

- Discussion on the reliability and validity of the results obtained from the system.
- Consideration of any potential biases or uncertainties in the data or analysis process.
- Evaluation of the generalizability of the results to real-world scenarios and different datasets.

4.1.5.Recommendations:

- Suggestions for potential improvements or enhancements to the MLBot system based on the findings and discussions.
- Recommendations for incorporating additional features, refining the algorithms, or expanding the dataset to enhance the system's performance.
- Consider ethical and legal implications related to the system's implementation and recommendations.

The section titled "Results and Discussion" offers a thorough examination and explanation of how well the MLBot system functions in identifying and stopping money laundering activities. This analysis is valuable in enhancing our comprehension of the system's effectiveness and serves as a foundation for future research and development aimed at improving its capabilities and addressing any limitations that have been identified.

4.4. CONCLUSION

All in all, "MLBot: A Simulated Intelligence Fueled Conditional Organization and Conduct Examination to Recognize and Forestall Tax evasion Exercises utilizing LSTM" is a refined framework intended to address the difficulties of identifying and forestalling tax evasion exercises in financial exchanges. The framework uses LSTM (Long Momentary Memory) brain organizations, alongside different information pre-handling and element designing procedures, to accomplish the exact grouping of exchanges into authentic and potential tax evasion classifications. The exhibition of MLBot is assessed utilizing different assessment measurements like exactness, accuracy, review, and F1-score. Disarray grid examination gives a point-by-point breakdown of genuine upsides, genuine negatives, bogus upsides, and misleading negatives, assisting with evaluating the framework's exhibition in identifying tax evasion exercises. The outcomes and conversation module gives a top-to-bottom investigation of the framework's exhibition, including the translation of discoveries, recognizable proof of impediments, and development proposals. This examination fills in as a reason for additional innovative work to upgrade the framework's capacities and address any distinguished difficulties. In synopsis, "MLBot: A computer-based intelligence Controlled Conditional Organization and Conduct Examination to Distinguish and Forestall Illegal tax avoidance Exercises utilizing LSTM" exhibits promising potential in really identifying and forestalling tax evasion exercises in financial exchanges. By utilizing progressed computer-based intelligence procedures and vigorous information investigation, the framework adds to fortifying the endeavours to battle monetary wrongdoings and keep up with the respectability of the financial framework.