Grenze International Journal of Engineering and Technology, January Issue

A Hybrid Modelling Approach for Detecting Money Laundering in Banking Sectors

M.D Jahan Khatoon[1], G. Nandhini Reddy[2], Shyam Kumar[3] and Dr. M Purushotham Reddy[4]
[1-3]Student, Institute of Aeronautical Engineering, Hyderabad, India
Email: jahan.md2304@gmail.com, gouninandhinireddy23@gmail.com, shyamkumarsudha2@gmail.com
[4]Professor and Head, Institute of Aeronautical Engineering, Hyderabad, India
Email: dr.purushothamreddy@iare.ac.in

***Abstract*—Money laundering poses a major challenge in the banking industry, as it allows criminals to mask illegal funds as legitimate income. Identifying these illicit financial activities is complex due to the massive volume of transactions and the constant evolution of laundering techniques. Traditional detection methods often fall short in uncovering sophisticated schemes, emphasizing the need for advanced machine learning techniques. This study examines the impact of the Random Forest and Decision Tree algorithms in identifying money laundering activities. Both algorithms are popular for classification, with Decision Tree known for its interpretability and Random Forest for its higher accuracy due to its ensemble structure These models can improve the detection process by examining huge datasets and discovering transaction patterns.We assess their performance using metrics like accuracy, precision, and recall. Additionally, this paper discusses challenges in applying machine learning to anti- money laundering (AML) initiatives, including issues like data imbalance and feature selection. The findings indicate that Random Forest, with its resilience and capacity to handle noisy data, often outperforms Decision Tree in identifying suspicious transactions. The research concludes with suggestions for implementing these techniques in banking systems and outlines future research directions to improve AML models through hybrid approaches.**

***Keywords*— Machine learning; Artificial intelligence; Anti-money laundering; Banks; Random Forest; Decision Tree.**

I. INTRODUCTION

Money laundering remains a critical issue for financial institutions globally, facilitating the transfer of illegal funds and supporting various criminal operations. Banks, as primary channels for financial transactions, are especially at risk from this threat. Many banks are using cutting-edge data analysis methods, such as machine learning, to address these issues and improve their ability to identify and stop money laundering. Two widely used machine learning models— Random Forest and Decision Tree algorithms—have shown effectiveness in identifying suspicious transaction patterns within large volumes of banking data.

The Decision Tree algorithm is a well-established method that works by creating a structured, tree-like model of conditional decisions, where each branch signifies a decision or classification determined by a specific feature in the dataset. In detecting money laundering, a Decision Tree can be trained on historical data to classify transactions as either normal or suspicious. Its interpretability and straightforward structure make Decision Trees valuable for identifying the factors that may signal suspicious financial behavior.

Grenze ID: 01.GIJET.11.1.225_1

However, Decision trees can sometimes overfit complex or noisy datasets, such as financial transaction records, meaning they may perform effectively on training data but have trouble adapting to new data.
To address overfitting, the Random Forest algorithm offers a robust alternative. To provide a final forecast, Random Forest, an ensemble learning technique, builds several decision trees and combines their results. Prediction accuracy is increased and overfitting is less likely using this ensemble approach. For detecting money laundering, Random Forest has several benefits over traditional methods. First, it effectively handles large datasets with numerous features, which is essential when analyzing the complex transaction records produced by banks. Second, it can model non-linear relationships between transaction features, allowing it to detect subtle patterns that may indicate suspicious activity. Additionally, Random Forest ranks feature importance, helping banks identify key indicators of unusual behavior, such as sudden transaction spikes or transfers involving multiple accounts. Once these models are trained, they can be implemented in real-time to monitor new transactions. When a transaction is flagged as suspicious, it can be reviewed further by compliance teams. Using algorithms such as Decision Tree and Random Forest in machine learning not only enhances the accuracy of detecting money laundering but also decreases the time required and resources required for manual transaction reviews, enabling banks to concentrate on high-risk cases and enhance operational efficiency.
The interpretability of Decision Tree and Random Forest models is another significant benefit for the financial sector, where transparency and accountability are crucial. Unlike some machine learning models, such as deep learning, these algorithms have decision processes that can be easily explained to regulators, which is essential for meeting antimoney laundering (AML) compliance requirements. For banks to maintain compliance with legal requirements, they must show how their systems identify questionable activity.
To sum up, machine learning techniques like Random Forest and Decision Tree offers promising solutions for identifying money laundering in the banking sector. These models enable the analysis of large, complex datasets to uncover suspicious patterns and improve the efficiency of AML efforts. Despite challenges like data imbalance and model interpretability, these algorithms' accuracy, scalability, and transparency make them valuable tools in the fight against financial crime. As banks continue to adopt machine learning, they will strengthen their ability to combat money laundering and adhere to regulatory standards.

II. Problem Statement

Money laundering is an ongoing and escalating threat within the global financial landscape, creating significant challenges for the integrity and stability of banking institutions. Criminal groups exploit banks to transfer illicit funds, often concealing their sources through complex transaction networks. This complexity makes it difficult for traditional rule-based systems to accurately detect money laundering activities. As these illegal schemes become increasingly sophisticated, advanced detection methods are essential. Consequently, the development of a strong and efficient system to identify suspicious transactions in real time has become a critical focus for both banks and regulatory bodies. The main obstacle is the enormous volume and intricacy of banking operations. Banks process millions of transactions each day across various regions, involving multiple currencies and account holders. Detecting money laundering patterns within this large dataset is a formidable task. Current rulebased systems, which rely on preset criteria like transaction size or country, often fall short because they cannot adapt to evolving laundering tactics. These systems also tend to generate high false-positive rates, overwhelming compliance teams and diverting resources from addressing actual criminal activity. This inefficiency underscores the pressing need for more flexible and intelligent detection solutions. A significant challenge in detecting money laundering is the imbalanced nature of banking data. Legitimate transactions significantly outnumber suspicious ones, which makes it difficult for machine learning models to effectively identify the rare instances of fraud. This class imbalance can result in models that favor the majority class, leading to missed detections of critical money laundering activities. To address this imbalance, careful tuning of machine learning algorithms is necessary. Techniques such as adjusting class weights, applying under-sampling or over-sampling methods, and using specialized metrics that highlight the significance of minority classes can be employed. Effectively tackling this issue is vital for enhancing the model's accuracy in identifying illicit activities. Ultimately, the aim is to create a smart, scalable solution capable of detecting money laundering in real time while minimizing false positives and boosting the efficiency of compliance teams within banks. By harnessing the strengths of both Random Forest and Decision Tree models, financial institutions can improve their monitoring of transactions for suspicious activities and ensure that financial crimes are identified more quickly and accurately. However, the successful development and implementation of such a system will require continuous refinement, as criminal organizations are constantly evolving their tactics to exploit vulnerabilities within the financial system.

III. DATASET

A. Data Collection

This project's data was sourced from a simulated dataset representing money laundering activities within banks in the Middle East, modeled on an authentic dataset. The simulated data closely resembles real transactions, utilizing similar features and data points to reflect actual transfer processes. Both money laundering and legitimate banking activities were included in the simulation, which was designed to mimic realworld scenarios comprehensively. The simulation mimics the three essential phases of money laundering: placement, layering, and integration. For every stage, specific rules were established for the inflow and outflow of funds, providing a detailed representation of laundering practices. A notable advantage of this synthetic dataset is its flexibility, allowing for adjustments and customization to generate data under various scenarios.

B. Feature Variables

Transaction type Transactions are categorized as either "cash-in" or "transfer-out." This categorical variable was converted into dummy variables to facilitate analysis. - Crime level: The crime level indicates whether the money laundering activities were carried out by the leader of a financial institution or by another employee within the same organization. This categorical variable was also transformed into dummy variables for modeling. - Transaction amount The transaction amount is represented as a continuous variable, reflecting the actual sum of money involved in each transaction. - Date The date reflects the specific day, month, and year a transaction occurred. Using Python's date and time functions, the date was reformatted into new categories, such as day of the week and month, to extract additional insights. - Time The transaction time is a continuous variable that is rounded to the closest hour.

C. Money Laundering Type

The stages of money laundering—placement, layering, and integration—were represented by coding each stage as a distinct category. These stages were then transformed into categorical variables for analysis. - Target variable: money laundering The target variable specifies whether a transaction was associated with money laundering. It was coded as "1" for transactions flagged as money laundering and "0" for those that were not. The target variable is represented as y = {1, if money laundering occurred 0, if no money laundering occurred}.The Fig 1. shows the class distribution.

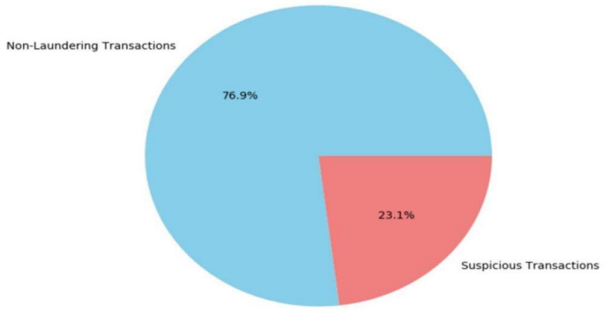

Fig 1. Class Distribution

D. Statistical Tools and Performance Metrics

For coding and analysis, the Python programming language was utilized within a Jupyter notebook environment. The Scikit-learn library, a prominent machine learning (ML) toolkit, was selected for developing and evaluating various ML algorithms.

To assess the effectiveness of the best classification algorithm on the test set, a confusion matrix was employed. Additionally, a classification report was generated that included performance indicators including the F1 score, recall, and precision. This report offered insights into the model's capacity to identify transactions associated with money laundering, in contrast to those that do not involve illicit activities.

E. Data Preprocessing

The data preprocessing stage encompassed multiple steps aimed at cleaning and preparing the dataset for analysis. Initially, we checked for any missing values within the data and subsequently eliminated any duplicate records. For each feature, we identified and documented the unique values present. When encountering Not a

Number (NaN) entries, we substituted them with zeros. Additionally, we determined the sender's location and assessed the minimum and maximum transaction amounts.

F. Feature Engineering

During the data preprocessing phase, several variables were modified to enhance their clarity. For instance, the variable "isfraud" was renamed to "moneylaundering" to indicate whether a transaction was associated with money laundering. Similarly, the coding for different types of money laundering was updated; "type1," "type2," and "type3" were reclassified as "placement," "layering," and "integration," respectively. To determine the five most important features from the dataset, feature selection techniques were used which contained a total of 2,340 observations. The data was subsequently split into a training set comprising 60% and a testing set making up 40%.To normalize the numerical features, a standard scaler technique was applied. Since raw date and time data could not be directly analyzed by the model, these were initially transformed into categorical variables for Exploratory Data Analysis (EDA) and later reverted to numerical format for analysis using machine learning and artificial neural network algorithms. Additionally, the time data was segmented into three categorical variables to represent morning, afternoon, and evening periods. The Fig. 2 shows the comparision of transactions.

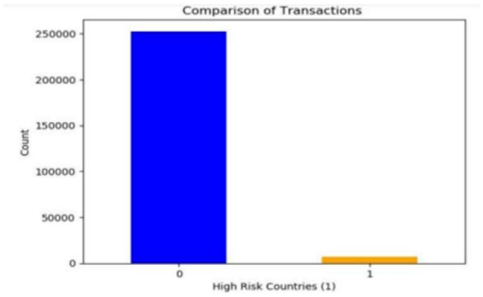

Fig 2. Comparision of Transactions

IV. Methodology and Implementation

A. Random Forest

Random Forest is a supervised machine learning algorithm that utilizes a collection of decision tree models for classification and prediction tasks. Each individual decision tree acts as a weak learner, possessing limited predictive power. This algorithm is grounded in ensemble learning, which combines multiple decision tree classifiers to enhance the model's accuracy. The Random Forest employs a bagging technique to create a diverse set of decision trees.

Given a dataset (X,Y) consisting of N total observations, where X represents the predictor variables and Y is the outcome variable, the Random Forest algorithm begins by generating Ki random variables for each observation i (where i=1,2,…,N). Each Ki random vector is subsequently converted into a decision tree, producing a collection of decision trees dKi(X) for all observations. The final classification is determined by the following expression:

$$D(X)=\text{argmax}\ \{\sum N\ dki(X) = Fraud, \sum N\ dki(X) = Not\ Fraud\}$$

One of the advantages of Random Forest is that it typically does not require a separate feature selection process. However, a limitation of this method is its potential to misclassify data with a wide range of values, particularly when variables have numerous categories. Random Forest is considered one of the most effective algorithms for fraud detection in the financial sector. Since uncertainty can arise during the initial stages of tree construction, it is essential to identify the most relevant features for analysis, especially when making decisions about node splitting. Fig 3. illustrates a random forest algorithm technique.

B. Decision Trees

Decision trees are non-parametric supervised learning methods that are commonly used for classification tasks. They create decision rules in a tree-like structure based on actual data attributes. The concept of decision trees is inspired by the way humans approach decision-making. Visually, they present information in a clear and intuitive tree format. A decision tree is composed of nodes, edges, and leaf nodes. It includes a series of branches or nodes that are connected by edges. The root node of the tree selects a feature that divides the data into two or more sub-nodes, leading to the creation of decision nodes. Following this initial partitioning, each sub node is

further divided into additional sub-nodes. This process continues iteratively until all training samples are accounted for, resulting in leaf nodes that represent the classifications of the data. Each leaf node ultimately indicates the predicted class for the observations that reach that point in the tree. Fig.4 shows the flow diagram of a decision tree.

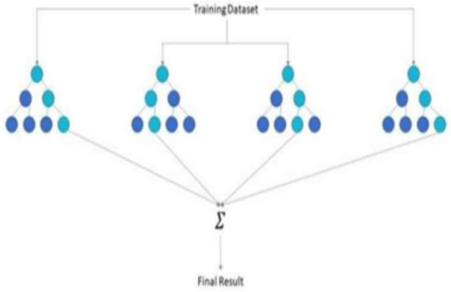

Fig 3. Random Forest

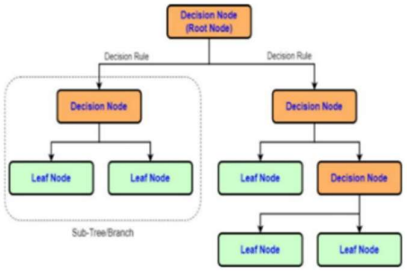

Fig 4. Decision Tree

C. Synthetic Minority Oversampling Technique

The Synthetic Minority Over-sampling Technique, or SMOTE, is a popular technique for addressing dataset class imbalance, especially in machine learning. When there is a large difference in the number of examples between classes — with one class (the minority) having significantly fewer instances than another (the majority) — this can lead to biased models. This technique addresses this by identifying samples within the minority class and concentrating on it, and then locating each sample's k-nearest neighbors. It then creates new, synthetic data points by interpolating between each sample and its neighbors, resulting in new examples that resemble, but aren't identical to, the original ones. This synthetic data effectively rebalances the dataset, enabling machine learning models to better learn patterns in the minority class and enhancing both model accuracy and fairness. SMOTE is especially beneficial in applications like fraud detection and medical diagnosis, where the occurrence of one class is naturally much lower than others.

D. Confusion Matrix

An extensive overview of a machine learning model's performance when assessed on a particular set of test data is provided by a confusion matrix. It displays the number of accurate and inaccurate predictions the model made, particularly in classification jobs where a categorical label is supplied to each input instance. The matrix provides precious perceptivity into the results produced by the model on the test data:

- True Pros (TP) The count of cases where the model rightly identifies a positive data point.
- True Cons (TN) The count of cases where the model rightly identifies a negative data point.
- False Pros (FP) The count of cases where the model inaptly predicts a positive data point.
- False Cons (FN) The count of cases where the model inaptly predicts a negative data point.

Using a confusion matrix is pivotal for assessing the performance of any bracket model because it provides a comprehensive breakdown of true pros, true cons, false pros, and false cons. This breakdown enables a more in-depth understanding of important criteria similar as recall, delicacy, perfection, and the model's overall capability to separate between classes. The confusion matrix is particularly precious when dealing with datasets that parade class distribution imbalances. An example of a confusionmatrix is illustrated in Fig 5.

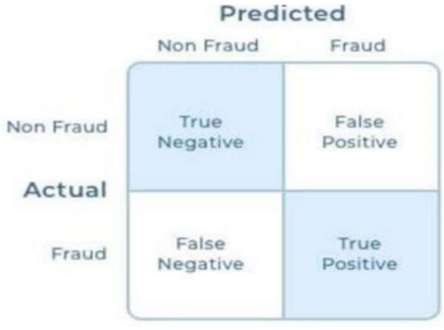

Fig 5. Example of Confusion Matrix

After testing a machine learning model, it's important to evaluate its performance by assessing its effectiveness and reliability. Four essential metrics for evaluating a model's performance are Accuracy, Precision, Recall, and F1 Score. Below, we will discuss each of these metrics along with their formulas.

To assess the model's delicacy, we use a confusion matrix, which is an NxN matrix that offers precious perceptivity into the model's performance regarding the bracket task. The confusion matrix consists of four crucial factors True Pros (TP), True Cons (TN), False Pros (FP), and False Cons (FN).

- Accuracy: The confusion matrix, which represents the general correctness of the classifications the model generates, is used to calculate a machine learning model's accuracy. The formula for accuracy: Accuracy = {TP + TN}/{TP + FP + FN + TN} - The ratio of accurate predictions to all predictions is determined by this formula. Accuracy values range from 0.0 (indicating no correct predictions) to 1.0 (indicating perfect accuracy).
- Precision: Precision assesses how accurate the model's positive predictions are. A higher precision value signifies increased trustworthiness in the model's initial predictions. Precision is the ratio of true pros predictions to the total pros predictions, encompassing both true pros and false pros. Precision = {TP}/{TP + FP} In this context, true pros (TP) represent instances where the model successfully identifies early-stage psoriasis, whereas false pros (FP) refer to cases in which the model mistakenly indicates that the condition is present.
- Recall: also referred to as acuity or the true positive rate, recall is a crucial metric for identifying early-stage psoriasis, as failing to detect positive cases (false cons) can have serious consequences. Recall is calculated as: Recall = {TP}/{TP + FN} In this formula, TP refers to the cases in which the model correctly identifies early-stage psoriasis, while FN denotes instances in which the model does not detect the condition when it is actually present.
- F1 Score: A metric called the F1 Score integrates precision and recall to evaluate a binary classification model's efficacy. When working with datasets that have an uneven distribution of classes, this statistic is quite helpful. The following formula is used to determine the F1 Score. F1 Score = 2 x {Precision x Recall}/{Precision + Recall} This metric offers an overall evaluation of a model's effectiveness, varying from 0 to 1, with higher values signify improved precision and recall. It is particularly important when considering both false pros and false cons.

V. Conclusion and Future Scope

In summary, Random Forest and Decision Tree algorithms have shown considerable effectiveness in identifying money laundering activities within the banking industry. Their capability to process large volumes of data, classify intricate transaction patterns, and support transparent decision-making processes makes them well-suited for combating fraud in the financial sector.

These models stand out for their predictive accuracy by harnessing the strengths of ensemble learning and hierarchical decision structures. As financial institutions confront increasingly sophisticated money laundering tactics, these machine learning tools provide a promising avenue for enhancing regulatory compliance and preserving the integrity of global financial systems.

Applying machine learning to detect money laundering represents a significant advancement for the banking sector. By adopting proactive strategies and employing these advanced algorithms, banks can improve their capacity to recognize and stop illicit financial activity.

As machine learning models evolve and regulatory frameworks become more robust, there is great potential for bolstering anti-money laundering efforts, improving operational efficiency, and promoting transparency in financial transactions.

In the near future, banks are poised to benefit from the integration of these models with high-speed processing systems that can analyze massive transaction volumes in real time. This advancement will enable instant identification of suspicious activities, allowing for more proactive prevention measures. Furthermore, stronger collaboration between banks and regulatory agencies could lead to shared intelligence databases, providing machine learning algorithms with access to larger datasets, which could significantly improve model reliability and detection accuracy across the banking.

Looking ahead, the role of these models in combating money laundering could be further strengthened by integrating additional technologies like blockchain, artificial intelligence, and deep learning. By combining multiple algorithms with the transparent ledger capabilities of blockchain, transaction traceability could be enhanced, making it even harder for illicit activities to go undetected. Decision tree algorithms could also evolve to work alongside neural networks, resulting in hybrid models that more effectively capture the nonlinear and complex patterns characteristic of money laundering. As these advancements continue, the banking sector will be better positioned to anticipate and counteract financial crimes on a global scale.

REFERENCES

[1] United Nations Office on Drugs and Crime. (2011). Estimating Illicit Financial Flows Resulting from Drug Trafficking and Other Transnational Organized Crimes.Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32.

[2] Schott, P. (2020). The Impacts of Money Laundering on the Global Economy: A Study. International Journal of Financial Studies, 8(3), 1-12

[3] Zhang, Y., Chen, L., & Gao, H. (2022). Comparative Analysis of Machine Learning Techniques for Financial Fraud Detection. Journal of Financial Services Research, 61(1), 123- 144

[4] Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. IBM Journal of research and development. 3(3), 210-229.Gonzalez, R., & Martinez, R. (2018). Assessing the performance of decision trees in detecting financial fraud. Expert Systems with Applications, 96, 178-189.

[5] Kumar, R., & Singh, A. (2020). Machine Learning for Financial Fraud Detection: A Review. Journal of King Saud University-Computer and Information Sciences.

[6] Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1986). Classification and Regression Trees. Wadsworth

[7] Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5-32Li.

[8] Patel, M., & Shah, D. (2021). Enhancing Financial Fraud Detection Using Machine Learning Techniques. International Journal of Computer Applications.

[9] Liaw, A., & Wiener, M. (2002). Classification and Regression by randomForest. R News, 2(3), 18-22.Smith, J., Lee, K., & Chan, W. (2021). Successful Implementation of Random Forest for AML Detection in Banking. Journal of Financial Crime.

[10] Zhang, Y., Chen, L., & Gao, H. (2022). Comparative Analysis of Machine Learning Techniques for Financial Fraud Detection. Journal of Financial Services Research, 61(1), 123- 144.

[11] Krawczyk, B. (2016). Learning from Imbalanced Data: Open Challenges and Future Directions. Progress in Artificial Intelligence, 5(4), 221-232.

[12] Gupta, P., Verma, R., & Gupta, S. (2023). A Hybrid Approach to Money Laundering Detection: Random Forest and Decision Trees. Journal of Financial Crime [11] Krawczyk, B. (2016). Learning from Imbalanced Data: Open Challenges and Future Directions. Progress in Artificial Intelligence, 5(4), 221-232.

[13] Ngai, E. W. T., Hu, Y., Wong, Y. H., Chen, Y., & Sun, X. (2011). The Application of Data Mining Techniques in Financial Fraud Detection: A Classification Framework. Decision Support Systems, 50(3), 559-569.

[14] Smith, J., Lee, K., & Chan, W. (2021). Successful Implementation of Random Forest for AML Detection in Banking. Journal of Financial Crime.

[15] Johnson, A., Smith, J., & Taylor, R. (2022). Decision Tree Analysis in Anti-Money Laundering: A Case Study. Journal of Money Laundering Control.

[16] Obermeyer, Z., Powers, B., Wang, J., et al. (2019). Dissecting Racial Bias in an Algorithm Used to Manage the Health of Populations. Science, 366(6464), 447-453.

[17] Huang, X.., Chen, X., &Zhang, W.(2023).The Role of Artificial Intelligence in Money Laundering Detection: A Review. Computers & Security, 116