International Journal of Engineering Research & Technology (IJERT)

Income Tax Fraud Detection using AI&ML: Insights from Four Algorithms

[1] Dr.Udayakumar N L, [2]Chandu P, [3]Darshan K, [4]Dhanush K R, [5]Dheeraj S

[1] Professor,CSE, SIET, Tumkur

[2] chandupchandu56@gmail.com, SIET, Tumkur

[3] darshvfg@gmail.com, SIET, Tumkur

[4] dhanushdanu083@gmail.com, SIET, Tumkur

[5] seenappadheeraj@gmail.com SIET, Tumkur

Abstract:

Tax fraud is one of the major issues faced by governments around the globe. It can be defined as an alteration of tax information to reduce liability. This fraud can be committed by increasing purchase values or reducing sales values. According to recent studies, the Indian government's losses from gold smuggling were estimated at around $1.6 billion. Countries like the United States, France, and Serbia use unsupervised machine learning for income tax fraud detection. A significant drawback of unsupervised learning models is their lack of interpretability and validation due to the absence of labelled data. Since the training of the model is done with unlabelled data, it becomes challenging to validate the correctness of the output. Additionally, it is hard to measure the accuracy and effectiveness of unsupervised models. They are also limited in their use when deployed independently for income tax fraud detection. The work done in this paper focuses on addressing the limitations of unsupervised learning models by involving four supervised machine learning models and analysing the accuracies and classification reports of each model. This involves using a dataset with more than two lakh entries and 17 columns.

Core technologies:

The methodologies used in this study include classifiers such as Random Forest, Decision Tree, Logistic Regression, and Support Vector Machine (SVM). To tackle the problem of class imbalance, we employ the Synthetic Minority Oversampling Technique (SMOTE), which helps create a balanced dataset for more effective model training. The pre-processing stage involves feature engineering, which includes label encoding for categorical features, normalization, and feature scaling. Our analysis is based on a dataset of 200,000 entries, with 44,048 identified as fraudulent, resulting in an approximate fraud rate of 22%. We address challenges related to class imbalance and computational efficiency by implementing optimized data sampling strategies and hyperparameter tuning. In the models assessed, both Random Forest and Decision Tree classifiers stood out for their impressive performance, achieving high accuracy and F1-scores while also being efficient in terms of computation time.

Performance insights:

The Random Forest and Decision Tree models enhanced with SMOTE emerged as the top performers, recording impressive weighted F1-scores of 0.98 and 0.97, respectively. Both models excelled, especially in identifying non-fraudulent cases, with Random Forest having a slight edge over the Decision Tree. The Decision Tree remained a strong contender while being more efficient in terms of computation time.

On the other hand, Logistic Regression and SVM with SMOTE, although beneficial, did not perform as well. Logistic Regression reached a weighted F1-score of 0.91 but had difficulties with recall in detecting fraudulent cases. SVM fared a bit better, achieving a weighted F1-score of 0.93, effectively balancing precision and recall, but it demanded more computational resources compared to the Decision Tree.

Keywords: Machine Learning, Fraud Detection, Income Tax, Financial Security, Random Forest, Logistic regression, SVM, Decision Tree, Imbalanced Data, SMOTE.

1. Introduction:

1.1 Background:

Published by, www.ijert.org

ISSN: 2278-0181

Tax fraud is one of the most significant issues affecting government revenue worldwide. According to reports and studies from Indian Customs and the World Gold Council (WGC), an estimated $1.6 billion (₹12,000 crores) is lost annually due to gold smuggling and tax evasion in India. Studies by the National Institute of Public Finance and Policy (NIPFP) and the International Monetary Fund (IMF) or OECD estimate that 1.5% of India's GDP is lost to corporate tax avoidance and evasion. Additionally, the Indian Ministry of Finance and the Central Board of Indirect Taxes and Customs (CBIC) report annual losses of ₹50,000 crore to ₹1 lakh crore due to GST fraud.

There are two main strategies used by the tax authorities [1] such as rule-based systems and manual auditing. The first strategy rule-based systems involve multiple if else statements where each of the tax returns pass through. The tax return is considered as fraud if any of the fails certain number of conditions. The second strategy involves the auditor checking the tax return manually and detecting a fraud based on their experience. The first strategy requires a lot of time in building and maintaining the system. It may not work for the new fraud strategies. The issue with the second strategy is that the experience of the auditor is subjective and it is manually done. In the recent times, due to the improvement of the technology the approach of detecting a fraud has changed. And the approach adopted by some of the countries with the machine learning models can overcome the issues with the above mentioned two strategies and also can improve in the detection of new patterns or new fraud strategies [1]. Several works done on the tax fraud detection evolves around unsupervised learning, supervised learning [2], data analytics and natural language processing. Unsupervised models work on the unlabelled data which is not useful if they are used independently for the detection of the fraud. When it comes to the data analytics where the model completely relies on the dataset. If the data is inconsistent, incomplete or inaccurate then it can lead to the wrong or unreliable predictions. NLP models may struggle to fully understand the context of textual information in tax returns, such as subtle variations in phrasing or ambiguous language.

1.2 Objectives:

Our primary goal is to reduce the fraud activities by using supervised machine learning models like logistic regression, random forest, decision trees, support vector machine.
These models goal is to overcome the disadvantages of the traditional methods and improve the fraud detection by identifying the new patterns.
The rest of the paper is as follows; in section 2 we will discuss on the previous works with respect to the tax fraud detection with various methods. In section 3 we will describe our approach to find the fraud. In the section 4 we will discuss on the dataset, experimental setup, machine learning models and the results of the individual models. At last we will discuss on the conclusion and the future works.

2. Literature review:

2.1 Data Mining and Hybrid Approaches

The detection of the fraud in financial statements using machine learning and data mining [3], which involves the ensemble method to categorize the fraud in dataset and datamining techniques with machine learning to increase the accuracy. It will have a advantage of detecting fraud with the high accuracy. But it is complex to detect fraud for unlabelled data using this hybrid model of machine learning and data mining.

2.2 Blockchain-Based Approaches

The detection of fraud using block chain [4], which involves the supervised learning methods which are evaluated based on the accuracy. It is easy to implement and less time required for the fraud detection. But high accuracy in dataset with only limited number of features and entries.

2.3 Neural Networks based approach

The fraud detection using neural networks [5], which is deployed artificial neural network (ANN) for the fraud detection. It can be able to process big data and considers many features of the dataset for fraud detection. But model's accuracy reduced as the number of layers increased.

2.4 Reinforcement learning

The enhanced income tax fraud detection [6], which is done using machine learning with the utilization of the boosting algorithms in machine learning to detect the potential instances of income tax fraud. It is combined multiple weak learners to create a strong predictive model. But fine tuning is difficult which may result in overfitting.

2.5 Hybrid approach

The involvement of both supervised and unsupervised models with compliance score of every tax payer [7]. The framework uses the data in its totality, by allowing it to detect the new pattern of fraud. But utilizing the entire dataset without a proper validation may lead to overfitting

2.5 Unsupervised learning based approach

The tax fraud detection for under reporting declarations [8], here detection of potential fraudulent taxpayers using only unsupervised learning techniques. The discovery of the hidden patterns in the data without prior knowledge of fraud instances. But the absence of labelled data makes it challenging to validate it challenging to validate the effectiveness of the detected patterns or clusters.

3. Methodology:

3.1 Dataset description:

This study will use a dataset of 2,00,000 records with 17 features. These features helped us to bring out all kind of aspects within the happened transactions. Data has numerical columns which includes the information of total transaction done on each category and so on while providing a contextual detail.

The columns consists of the income declared by the tax payer, the revenue generated from the business, expenditure on living, spending on luxury items, online spending, tax paid on the property, maintenance of vehicle, salaries provided for the employees.
BUSINESS_REVENUE, LIVING_COST, LUXURY_SPENDING, ONLINE_SPENDING, PROPERTY_TAX, CAR_MAINTAINANCE, EMPOYEE_SALARY.

These are the inputs taken from the tax officer. Then the calculations for the feature engineering will happen. Like EXPENSES, INCOME_TO_EXPENSE_RATIO, ESTIMATED_INCOME, INCOME_DIFFERENCE, TOTAL EXPENDITURE, SPENDING_TO_INCOME_RATIO, HIGH_VALUE_PURCHASE_FLAG, EXCESSIVE_SPENDING_FLAG and the main target variable is binary label POTENTIAL_TAX_FRAUD indicating if the transactions are fraudulent (1) or not (0).

The dataset is imbalanced, with only 22% of fraudulent entries compared to non-fraudulent entries. There are some missing columns which were filled during pre-processing. However the data is enough to explore machine learning models for fraud detection.

The size of dataset is 22.63MB, which makes it good enough for experimenting with many models which offer complexity and which also able to handle large scale frauds. The following table shows few rows of dataset as presented.

business_revenue	living_cost	luxury_spending	online_spending
1948650	516174	97815	10077
925335	437697	26373	99933
1379331	537493	91862	67212
1958770	469804	32231	63741
1285409	679212	127192	21197
Expenses	**income_to_expense_ratio**	**estimated_income**	**income_differen**
888410	0.249837	389730	167772
821373	0.300554	185067	-61800
1098915	0.211055	275866.2	43934.2
1001580	0.465103	391754	-74084
1115275	0.322053	257081.8	-102096

excessive_spending_flag	potential_tax_fraud
1	1
1	0
1	0
1	0
1	0

3.2 Models used:

In this paper, 4 different models – Random forest, decision trees, logistic regression, support vector machine are used to detect the fraudulent cases. These models are used because of their proven efficiency and effectiveness in machine learning tasks and the ability to handle the complex entries of fraud detection in datasets.

- **Random forest:**
  Random Forest, a popular technique within ensemble learning, is often employed in the prediction modelling context with large and complex data sets. By taking the averaged votes of multiple decision trees, this approach not only reduces the risk of overfitting but also enhances the accuracy of the model. In terms of identifying parameters that predict students' achievement, such as history of academic performance, engagement, and socio-economic status, Random Forest proves to be more accurate. It also makes use of feature selection so that while making decisions, only the relevant information is included. The fact that Random Forest models have the capability to model non-

linear relations in the data set adds strength to the use of this model in predicting students' academic performance.

- **Decision tree:**

  A Decision Tree is like a flowchart that helps make decisions by answering a series of questions. Imagine it as a tree, where each branch represents a decision based on certain conditions, and each leaf shows the final outcome. The tree works by splitting the data into smaller and smaller groups at each node based on features that best separate the data. It's easy to understand because it's like asking a yes/no question at each stage. While it's a simple model, it can easily overfit, meaning it might become too tailored to the training data and not generalize well on new data. But, when used correctly, it's a great tool for both classification and regression tasks. Plus, it's easy to visualize and interpret, making it a popular choice for those who need a transparent model.

- **Support vector machine:**

  Support Vector Machine (SVM) is a powerful algorithm that works by finding the best boundary (or hyperplane) that separates data into different categories. Think of it as drawing a line between two groups of points in a 2D space in a way that maximizes the gap between them. This "margin" helps the model make more accurate predictions. SVM can handle complex data, even when it's not linearly separable, by using something called the kernel trick, which allows it to work in higher-dimensional spaces. It's especially good when there's a clear margin of separation between classes, but it can be a bit slow when dealing with large datasets. It also requires careful tuning to work at its best, but once it's set up, SVM can achieve impressive results.

- **Logistic regression:**

  Logistic Regression is a statistical model used to predict the probability of a binary outcome, like whether an email is spam or not, or whether a customer will buy a product. Despite its name, it's actually not about regression, but about classification. The model works by finding a line or curve (in higher dimensions) that best separates the two classes. It outputs a probability value between 0 and 1, which can be converted into a class label. It's simple and fast, making it ideal for many situations, but it assumes a linear relationship between the features and the outcome. It works well when the data has clear boundaries, but might struggle when the data is more complex or not linearly separable.

3.3 Balancing a data:

SMOTE (Synthetic Minority Over-sampling Technique)

SMOTE is a clever technique used to handle the problem of imbalanced data, where one class (usually the minority class) is underrepresented compared to the other class (usually the majority class). In simple terms, SMOTE helps give more "voice" to the minority class, so that the model doesn't just focus on predicting the majority class. Instead of simply duplicating minority class examples (which could lead to overfitting), SMOTE generates synthetic data points. It creates new, artificial samples by taking the features of existing minority class points and generating new ones that are similar, but slightly different. This is done by finding the nearest neighbours of each data point and creating new points in between. Think of it like filling in the gaps in a map where the minority class is sparsely located. By doing so, SMOTE helps create a more balanced dataset, allowing the model to better understand the minority class and improve prediction performance. It's especially useful when dealing with rare events, like fraud detection, where the fraudulent cases are much fewer than the non-fraudulent ones.

3.4 Feature engineering:

3.4.1 Expenses:

expenses = living_cost + luxury_spending + online_spending + property_tax + car_maintenance + employee_salary

Total expenses are the sum of all the costs, such as living costs, luxury purchases, online spending, property taxes, car maintenance, and employee salaries. This helps to understand how much money is going out in various areas.

3.4.2 Income to expense ratio:

income_to_expense_ratio = income_declared / expenses if expenses != 0 else 0

This ratio compares the declared income to total expenses. If expenses are zero, the ratio is set to 0. A higher ratio indicates more income relative to spending.

3.4.3 Estimated income:

estimated_income = business_revenue * 0.20
This estimates the income based on business revenue, assuming 20% of the business revenue contributes to the declared income. It helps assess if the declared income matches typical business earnings.

3.4.4 Income Difference:

income_difference = estimated_income - income_declared
This shows the difference between the estimated income (based on revenue) and the declared income. A large difference may suggest underreporting or discrepancies in reported earnings.

3.4.5 Total Expenditure:

total_expenditure = living_cost + luxury_spending + online_spending
This is the sum of living costs, luxury spending, and online purchases, representing the core expenses related to personal lifestyle and non-business spending.

3.4.6 Spending to Income Ratio:

spending_to_income_ratio = total_expenditure / income_declared if income_declared != 0 else 0
This ratio compares total spending to declared income. If the ratio is high, it means a large portion of the income is being spent, which may indicate excessive spending.

3.4.7 High Value Purchase Flag:

high_value_purchase_flag = 1 if luxury_spending > 100000 else 0
If luxury spending exceeds 100,000, this flag is set to 1, indicating a high-value purchase. Otherwise, it's set to 0, showing no significant luxury spending.

3.4.8 Excessive Spending Flag:

excessive_spending_flag = 1 if spending_to_income_ratio > 0.8 else 0
This flag is set to 1 if the spending to income ratio is greater than 0.8, indicating that the person is spending more than 80% of their income, which is considered excessive. Otherwise, it's set to 0.

3.5 Evaluation metrics:

3.5.1 Accuracy:

$$Accuracy = (TP+TN) / TOTAL\ INSTANCES$$

Accuracy tells us how often the model is correct. It's the ratio of correct predictions (both fraudulent and non-fraudulent) to the total number of samples. While it gives a general idea of model performance, it can be misleading, especially when the dataset is imbalanced (e.g., more non-fraudulent cases than fraudulent ones).

3.5.2 Precision:

$$Precision=TP/(TP+FP)$$

Precision focuses on how many of the fraud cases predicted by the model are actually true frauds. In simple terms, it tells us how reliable the model is when it says a transaction is fraudulent. A higher precision means fewer false positives (incorrectly labeled frauds).

3.5.3 Recall:

$$Recall=TP/(TP+FN)$$

Recall, or sensitivity, tells us how many actual frauds were detected by the model. It measures the model's ability to identify as many fraudulent cases as possible. High recall is critical in fraud detection because we want to minimize the number of undetected fraud cases.

3.5.4 F1-Score:

$$F1\text{-}score=2((Precision\times Recall)/(Precision+Recall))$$

The F1-score combines precision and recall into one metric. It's useful in situations where there's a trade-off between precision and recall. If there's a large imbalance in the dataset, the F1-score gives us a balanced view of how well the model is detecting

fraud while minimizing false positives and false negatives.

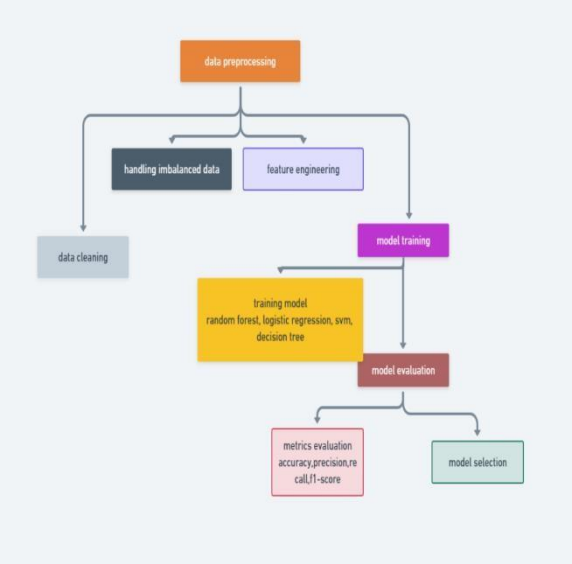

Fig 1. Flowchart illustrating the end-to-end process followed in the research:

4. Results and discussion:

This section explains the evaluation metrices of multiple machine learning models for detecting the fraud. The models include random forest, logistic regression, decision tree, svm.

4.1 Performance matrix comparison:

Model	Accuracy	Precision (Class 0)	Recall (Class 0)	F1-Score (Class 0)	Precision (Class 1)	Recall (Class 1)	F1-Score (Class 1)	Weighted Average F1-Score
Logistic Regression	0.9107	0.98	0.90	0.94	0.73	0.94	0.82	0.91
Random Forest	0.9801	0.98	0.99	0.99	0.98	0.93	0.95	0.98
Decision Tree	0.9788	0.98	0.99	0.99	0.97	0.93	0.95	0.98
Support Vector Machine	0.9233	0.98	0.92	0.95	0.76	0.94	0.84	0.93

Table 1. Performance Metrics for All Models

4.2 Visualization

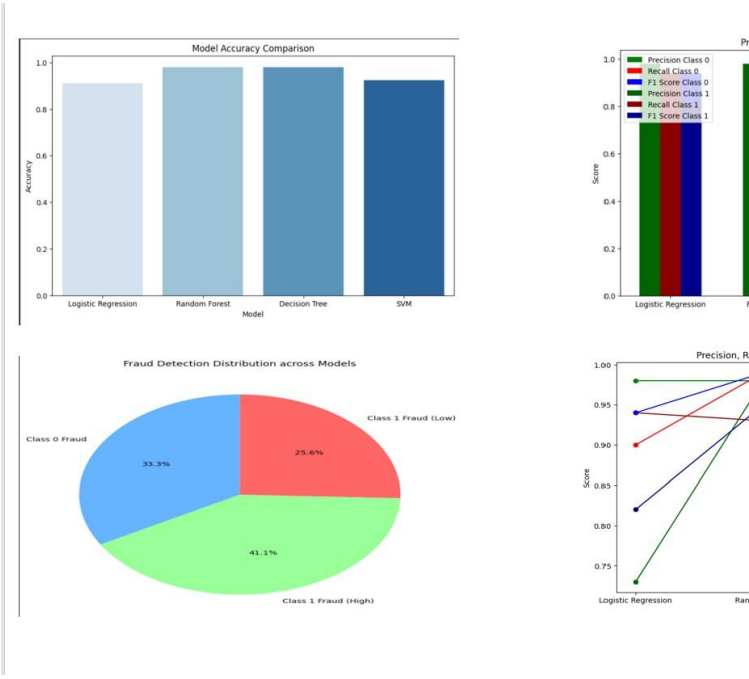

Figure 2. Model performances' evaluation graphs

4.3 Discussions:

In this study, I evaluated various machine learning models designed to detect income tax fraud. The researchers implemented feature engineering and SMOTE to address class imbalance in the dataset. The models assessed included Logistic Regression, Random Forest, Decision Tree, and Support Vector Machine (SVM). Each model was evaluated using metrics such as accuracy, precision, recall, F1-score, and computational efficiency. The

Random Forest model emerged as the top performer, achieving an impressive accuracy of 98.01%, a precision of 0.98, and an F1-score of 0.97. These results underscore Random Forest's capability to balance precision and recall, making it a robust choice for fraud detection, particularly in imbalanced datasets. However, its ensemble nature resulted in higher computational costs, with a processing time of 26.94 seconds, which could pose a challenge for real-time applications.

The Decision Tree model also showed strong performance, with an accuracy of 97.88%, precision of 0.97, and an F1-score of 0.95. While it was slightly less accurate than Random Forest, it was significantly faster, completing tasks in just 0.79 seconds. Despite its speed, Decision Tree models are more susceptible to overfitting, which may impact their performance on new, unseen data.

Logistic Regression, when combined with SMOTE, achieved an accuracy of 91.07%, demonstrating reasonable effectiveness for a simpler model. It recorded a precision of 0.73 and an F1-score of 0.82. Although Logistic Regression was able to identify a fair amount of fraud, its lower precision compared to the other models indicates a higher rate of false positives.

Support Vector Machine (SVM) attained an accuracy of 92.33%, with a precision of 0.76 and an F1-score of 0.84. SVM exhibited good recall but was less efficient than both Random Forest and Decision Tree in terms of computational time, requiring more time for training and prediction. All models benefited from feature engineering and the application of SMOTE, which helped to mitigate class imbalance.

5. Conclusion and Future Work:

5.1 Conclusion

This project focused on using different machine learning models to detect income tax fraud. We experimented with four popular models: Logistic Regression**,** Random Forest**,** Decision Tree, and Support Vector Machine (SVM). By applying SMOTE to balance the dataset and doing some careful feature engineering, we aimed to improve the models' ability to predict fraud effectively.

After analyzing the results, Random Forest clearly came out as the winner. It achieved the highest accuracy and showed strong performance across all metrics, making it the best model for fraud detection in our case.

5.1.1 Best Performing Model

The Random Forest model stood out with an impressive accuracy of 98.01%**.** Not only did it correctly identify fraudulent cases, but it also maintained a good balance between catching frauds and avoiding false alarms. It was a robust and reliable model, making it ideal for fraud detection in real-world scenarios.

5.1.2 Other Contenders

Other models also performed well, though not quite as well as Random Forest. Decision Tree was a solid choice, achieving an accuracy of 97.88%**.** It's faster than Random Forest, which made it a good option when computational time is a concern. Logistic Regression and SVM were useful, but they didn't quite match the performance of the tree-based models, especially when it came to identifying fraudulent activities.

5.2 Limitations

Every model had its strengths and weaknesses, and we did face some limitations along the way:

1. **Class Imbalance**: The dataset had a very low fraud rate (around 0.9%), which is a common issue in fraud detection. Even though SMOTE helped balance things out, the model still struggled with this imbalance, which affected the accuracy of fraud detection.
2. **Computational Costs**: While Random Forest performed excellently, it can be heavy on resources. This might be a problem when dealing with very large datasets, where training time could become an issue.
3. **Overfitting**: The Decision Tree model performed well but was prone to overfitting, which made it less reliable on

new, unseen data. This is a common challenge with decision trees when they are not properly tuned.

4. **Feature Engineering**: Although the feature engineering process improved the models, there's always room for improvement. Better, domain-specific features might help boost model performance even further.

5.3 Future Work

There are several exciting directions for future work that could make fraud detection even more effective:

1. **Addressing Class Imbalance Better**: We can explore more advanced techniques for balancing the dataset, like generating synthetic fraud cases using methods like SMOTE or even more specialized oversampling techniques.
2. **Hybrid Models**: A great next step could be combining the strengths of different models. For example, blending tree-based models like Random Forest with deep learning models could improve both accuracy and the ability to detect complex fraud patterns.
3. **Improving Features**: We could dive deeper into the data and extract more meaningful features. Better feature engineering, especially using domain-specific knowledge, could make a big difference in boosting model performance.
4. **Real-Time Fraud Detection**: Another important area for improvement is ensuring that these models can work in real-time. Fraud detection needs to be fast, especially in high-volume, real-time environments. Optimizing models for speed while maintaining accuracy is crucial for practical use.
5. **Making Models More Interpretable**: One area that could be improved is making these models more understandable. Tools like SHAP (SHapley Additive exPlanations) can help us explain why a model makes certain predictions, which could be useful for building trust and making decisions based on the model's results.

5.4. Summary

To sum up, this project successfully compared several machine learning models for detecting income tax fraud. Random Forest emerged as the best performer, providing high accuracy and a good balance between precision and recall. Decision Tree was also a good option, especially for faster, more interpretable results, though it didn't match the accuracy of Random Forest**.** SVM and Logistic Regression showed useful results but weren't as effective in identifying fraud. The use of SMOTE and feature engineering helped improve performance, especially in handling the class imbalance. However, challenges like overfitting, computational cost, and dataset imbalance remain. In the future, working on these issues and further optimizing models will lead to even better fraud detection systems.

References:

[1] A. Cobham and P. Jansky, "Global distribution of revenue loss from corporate tax avoidance: Re-estimation and country results," Journal of International Development, 2018.

[2] E. Crivelli, R. A. De Mooij and M. Keen, "Base erosion, profit shifting and developing countries," FinanzArchiv/ Public Finance Analysis, 2016.

[3] Matin N. Ashtiani, Bijan Raahemi, "Intelligent Fraud Detection in Financial Statements Using Machine Learning and Data Mining: A Systematic Literature Review",2021

[4] Rohan Kumar C L, Ali Mohammed Zain, Sanjay Kumar H P, Prajwal A V, Dr. Sudarshan R , "Comparative Study of Machine Learning Algorithms for Fraud Detection in Blockchain", 2022

[5] Belle Fille Murorunkwere, Origene Tuyishimire, "Dominique Haughton Fraud Detection Using Neural Networks: A Case Study of Income Tax.", 2022

[6] Dr RM Rani, Amrit Anand, Pratham Agarwal, "Enhanced Income Tax Fraud Detection System Using Machine Learning.",2024

[7] N. Alsadhan, "A Multi-Module Machine Learning Approach to Detect Tax Fraud.", 2022

[8] Daniel de Roux, Boris Perez, Andrés Moreno, Maria del Pilar Villamil, César Figueroa, "Tax Fraud Detection for Under-Reporting Declarations Using an Unsupervised Machine Learning Approach.", 2018