

# **Income Tax Fraud Detection Using AI&ML**

## **A PROJECT REPORT**

*Submitted by,*

<b>Mr. Mohan R Shetty</b>	<b>20211CSE0591</b>
<b>Ms. Amulya S Sathish</b>	<b>20211CSE0584</b>
<b>Ms. Suchithra K</b>	<b>20211CSE0599</b>
<b>Mr. Sumanth R</b>	<b>20211CSE0607</b>

*Under the guidance of,*

**Ms. Vineetha B**

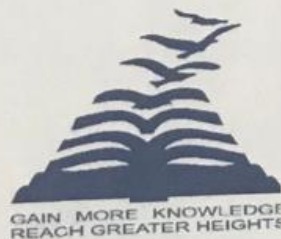
*in partial fulfillment for the award of the degree of*

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING, COMPUTER ENGINEERING,  
INFORMATION SCIENCE AND ENGINEERING Etc.**

**At**



**PRESIDENCY UNIVERSITY**

**BENGALURU**

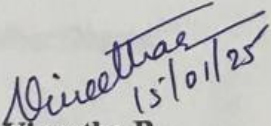
**DECEMBER 2024**

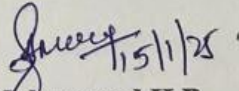
# PRESIDENCY UNIVERSITY

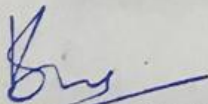
## SCHOOL OF COMPUTER SCIENCE ENGINEERING

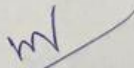
### CERTIFICATE

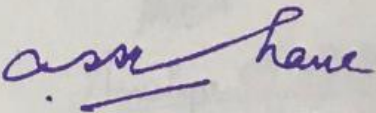
This is to certify that the Project report “**Income Tax Fraud Detection Using Ai&ML**” being submitted by “Mohan R Shetty, Amulya S Sathish, Suchithra K, Sumanth R” bearing roll number(s) “20211CSE0591, 20211CSE0584, 20211CSE0599, 20211CSE0607” in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering is a bonafide work carried out under my supervision.

  
**Ms. Vineetha B**  
Assistant Professor  
School of CSE&IS  
Presidency University

  
**Dr. Asif Mohammed H.B**  
Associate Professor & HoD  
School of CSE&IS  
Presidency University

  
**Dr. L. SHAKKEERA**  
Associate Dean  
School of CSE  
Presidency University

  
**Dr. MYDHILI NAIR**  
Associate Dean  
School of CSE  
Presidency University

  
**Dr. SAMEERUDDIN KHAN**  
Pro-VC School of Engineering  
Dean -School of CSE&IS  
Presidency University


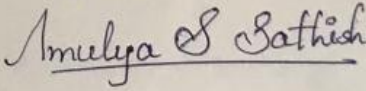
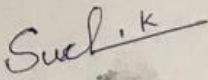
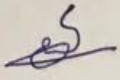
## PRESIDENCY UNIVERSITY

### SCHOOL OF COMPUTER SCIENCE ENGINEERING

#### DECLARATION

We hereby declare that the work, which is being presented in the project report entitled **Income Tax Fraud Detection Using AI & ML** in partial fulfillment for the award of Degree of **Bachelor of Technology in Computer Science and Engineering**, is a record of our own investigations carried under the guidance of **Ms.Vineetha B, Assistant Professor, School of Computer Science Engineering & Information Science, Presidency University, Bengaluru.**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

Student Name	Roll Number	Signature
Mohan R Shetty	20211CSE0591	
Amulya S Sathish	20211CSE0584	
Suchithra K	20211CSE0599	
Sumanth R	20211CSE0607	



## ABSTRACT

Income tax fraud poses a significant challenge to governments worldwide, undermining revenue collection. This project leverages Artificial Intelligence (AI) and Machine Learning (ML) techniques to build a robust system capable of detecting tax fraud with high accuracy and efficiency. By harnessing diverse income tax datasets and focusing on algorithms such as Random Forest algorithm, the methodology centres on comprehensive data collection, encompassing attributes crucial for fraud detection, serving as the bedrock for training the algorithms. The core of the project is the implementation of the Random Forest Algorithm, an ensemble learning method known for its robustness and ability to handle complex classification problems. The algorithm analyzes patterns in taxpayer data, including income, deductions, expenses, and tax payments, to identify anomalies and predict potential fraud cases. By comparing actual tax paid with expected tax based on predefined tax slabs, the system flags discrepancies that might indicate fraudulent activities. The model was trained on a comprehensive dataset containing taxpayer financial records, including features such as income, deductions, and expenses. Through feature engineering, a crucial variable expected tax was computed to enhance the predictive power of the model. The system achieved high accuracy in detecting fraudulent cases, demonstrating the effectiveness of Random Forest in classification tasks. Ultimately, the proposed system seeks to automate income tax fraud detection, streamlining processes, improving efficiency, and bolstering the security of tax assessments, thereby contributing to a more robust and effective tax system.

## ACKNOWLEDGEMENT

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Pro-VC, School of Engineering and Dean, School of Computer Science Engineering & Information Science, Presidency University for getting us permission to undergo the project.

We express our heartfelt gratitude to our beloved Associate Deans **Dr. Shakkeera L and Dr. Mydhili Nair**, School of Computer Science Engineering & Information Science, Presidency University, and **Dr. Asif Mohammed H.B**, Head of the Department, School of Computer Science Engineering & Information Science, Presidency University, for rendering timely help in completing this project successfully.

We are greatly indebted to our guide **Ms. Vineetha B, Assistant Professor** and Reviewer **Dr. N Thrimoorthy, Assistant Professor**, School of Computer Science Engineering & Information Science, Presidency University for her inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the project work.

We would like to convey our gratitude and heartfelt thanks to the PIP2001 Capstone Project Coordinators **Dr. Sampath A K, Dr. Abdul Khadar A and Mr. Md Zia Ur Rahman**, department Project Coordinators **Mr. Amarnath J.L** and Git hub coordinator **Mr. Muthuraj**.

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

**Mohan R Shetty**

**Amulya S Sathish**

**Suchithra K**

**Sumanth R**

## LIST OF FIGURES

Sl. No.	Figure Name	Caption	Page No.
1	Figure 4.1	Tax dataset	15
2	Figure 6.1	Architecture	20
	Figure 6.2	Flowchart	22
	Figure 6.3	Database Schema	26
3	Figure 7.1	Timeline of the project	30
4	Figure 9.1	Accuracy plot representation	34
	Figure 9.2	Precision plot representation	35
	Figure 9.3	Recall plot representation	36
	Figure 9.4	F1 Score plot representation	36
	Figure 9.5	Confusion Matrix Values	37
5	Figure 12.1	Initialization of flask web application	44
	Figure 12.2	Frontend of the model	44
	Figure 12.3	Output	45
	Figure 12.4	Values of the evaluation metrics	45
	Figure 12.5	Representation of the evaluation metrics	46
	Figure 12.6	Pictorial representation of the metrics	47
	Figure 12.7	Database Entries	47

# **TABLE OF CONTENTS**

<b>CHAPTER NO.</b>	<b>TITLE</b>	<b>PAGE NO.</b>
	<b>ABSTRACT</b>	<b>i</b>
	<b>ACKNOWLEDGMENT</b>	<b>ii</b>
<b>1.</b>	<b>INTRODUCTION</b>	<b>1</b>
	1.1 Overview of Income Tax Fraud	1
	1.2 Role of AI and ML in Fraud Detection	2
	1.3 Focus on Random Forest Algorithm	3
<b>2.</b>	<b>LITERATURE REVIEW</b>	<b>5</b>
	2.1. Existing method advantages and disadvantages	8
<b>3.</b>	<b>RESEARCH GAPS OF EXISTING METHODS</b>	<b>10</b>
<b>4.</b>	<b>PROPOSED MOTHODOLOGY</b>	<b>14</b>
	4.1 Algorithm Selection	14
	4.2 Data Collection and Preprocessing	15
	4.3 Feature Engineering	17
<b>5.</b>	<b>OBJECTIVES</b>	<b>19</b>
<b>6.</b>	<b>SYSTEM DESIGN &amp; IMPLEMENTATION</b>	<b>20</b>
	6.1 System Architecture	20
	6.2 Software and Hardware Requirements	22
	6.3 Frontend Design	23
	6.4 Backend Design	25
<b>7.</b>	<b>TIMELINE FOR EXECUTION OF PROJECT</b>	<b>30</b>
<b>8.</b>	<b>OUTCOMES</b>	<b>31</b>
	8.1 Evaluation Metrics	31
	8.2 Accuracy of Model Predictions	33
	8.3 Significance of Results	33
<b>9.</b>	<b>RESULTS AND DISCUSSIONS</b>	<b>34</b>
	9.1 Model Performance	34
	9.2 Comparison with Traditional Fraud Detection Methods	37

	9.3 Limitations	38
<b>10.</b>	<b>CONCLUSION</b>	<b>40</b>
<b>11.</b>	<b>REFERENCES</b>	<b>42</b>
<b>12.</b>	<b>APPENDIX-A</b> <b>PSEUDO CODE</b>	<b>44</b>
<b>13.</b>	<b>APPENDIX-B</b> <b>SCREENSHOTS</b> <b>APPENDIX-C</b> <b>ENCLOSURES</b>	<b>46</b>  <b>50</b>
	<b>1. Journal publication</b>	<b>50</b>
	<b>2. Plagiarism Report</b>	<b>51</b>
	<b>3. SDG Mapping</b>	<b>52</b>



# **CHAPTER-1**

## **INTRODUCTION**

### **1.1 Overview of Income Tax Fraud**

Income tax fraud poses a significant challenge to governments worldwide, leading to substantial revenue losses and undermining the fairness of the taxation system. Fraudulent activities can take various forms, including underreporting income, inflating deductions, falsifying tax credits, or failing to file returns altogether. These activities not only erode the trust of honest taxpayers but also reduce the funds available for public services such as education, healthcare, and infrastructure development. According to studies conducted by the Organization for Economic Co-operation and Development (OECD), tax fraud and evasion contribute to a substantial tax gap—the difference between the taxes owed and those collected. In some countries, this gap amounts to billions of dollars annually. Misreporting income to the Income Tax Department is a significant challenge faced by governments worldwide. This unethical practice, which includes underreporting income, overstating deductions, or falsifying financial records, directly impacts a nation's revenue generation and undermines the fairness of the taxation system. Individuals or entities engaging in such behavior aim to reduce their taxable income and evade paying their fair share of taxes, leading to a phenomenon commonly known as income tax fraud. Detecting such fraud is critical not only for increasing government revenue but also for maintaining the integrity of the tax system and ensuring equity among taxpayers. Traditionally, tax fraud detection relied on manual audits and rule-based systems. However, these methods are resource-intensive, time-consuming, and often ineffective in identifying sophisticated fraud schemes. Detecting misreported income is a complex task for tax authorities, as fraudulent activities often involve sophisticated schemes and the manipulation of financial data. Traditional methods, such as manual audits and rule-based systems, are resource-intensive and limited in their ability to identify complex patterns or adapt to new fraud techniques. With the exponential growth of data generated by taxpayers and the increasing complexity of fraudulent behavior, modern tax authorities require advanced tools capable of analyzing massive datasets efficiently and accurately. This has led to the integration of technology, particularly artificial intelligence (AI) and machine learning (ML), into fraud detection systems.

## **1.2 Role of AI and ML in Fraud Detection**

AI and ML have revolutionized the way financial systems operate, particularly in areas requiring pattern recognition and anomaly detection. In the context of tax fraud detection, these technologies offer powerful capabilities for analyzing complex datasets, identifying suspicious patterns, and predicting fraudulent activities. Unlike traditional rule-based systems that rely on predefined conditions, AI and ML models can learn from historical data, adapt to evolving fraud tactics, and uncover hidden relationships within the data. Machine learning models, in particular, excel in processing large volumes of structured and unstructured data to identify irregularities. For example:

- **Classification Algorithms:** These models categorize taxpayers into groups such as "high-risk" or "low-risk," enabling tax authorities to prioritize audits effectively.
- **Anomaly Detection:** ML algorithms can identify outliers in data, such as unusually low tax payments relative to reported income or deductions that exceed expected thresholds.
- **Predictive Analytics:** By analyzing past data, ML models can predict the likelihood of fraud in new cases, helping tax authorities to pre-emptively address potential issues.

### **The Need for AI and ML in Fraud Detection**

- **Growing Complexity of Financial Data:**  
Modern financial systems generate vast amounts of structured and unstructured data, including tax returns, invoices, bank statements, and transaction logs. AI and ML can process and analyze these datasets at a scale and speed that manual methods cannot match.
- **Evolving Fraud Tactics:**  
Fraudsters continually develop sophisticated methods to bypass traditional detection systems. Static, rule-based approaches often fail to keep up with these evolving tactics. ML models, however, learn from historical fraud patterns and adapt over time, making them resilient to new strategies.
- **Limited Resources for Manual Audits:**  
Tax authorities face resource constraints that make it impossible to scrutinize every tax return. AI-driven systems prioritize high-risk cases for further investigation, optimizing the allocation of resources and increasing efficiency.

AI and ML systems also improve over time, as they are capable of continuously learning

from new data. This adaptability is particularly important in combating fraud, as perpetrators constantly develop new techniques to evade detection. Furthermore, ML models can handle data imbalances effectively, ensuring that rare instances of fraud are not overlooked. In addition to improving detection accuracy, AI-driven systems enhance efficiency by automating repetitive tasks and reducing the workload of human auditors. This allows tax authorities to allocate their resources more effectively, focusing on high-priority cases while minimizing false positives.

### **1.3 Focus on Random Forest Algorithm**

Libraries like scikit-learn. Hyperparameters such as the number of trees and tree depth are tuned for optimal performance, and cross-validation is employed to prevent overfitting. Model evaluation focuses on metrics like precision, recall, and F1-score, with an emphasis on recall to minimize false negatives, which is critical for fraud detection. Additionally, Random Forest helps identify the most important features that contribute to fraud prediction, aiding in the understanding of which taxpayer behaviors are red flags. Income tax fraud detection using AI/ML techniques, specifically the Random Forest algorithm, offers an effective approach to identifying fraudulent behavior in tax filings. The Random Forest is an ensemble learning method that constructs multiple decision trees and combines their predictions for improved accuracy and robustness. It excels in handling large datasets, reducing overfitting, and providing insights into feature importance, making it particularly suited for complex, high-dimensional data typical of tax filings. To build a fraud detection model, the data needs to be carefully prepared, which involves handling missing values, encoding categorical features, and balancing the dataset to address class imbalance between fraud and non-fraud cases. Once the data is preprocessed, the Random Forest model is trained using the RandomForestClassifier from however, challenges such as handling imbalanced data, performing effective feature selection, and interpreting individual predictions remain. Despite these challenges, Random Forest is highly effective for real-time fraud detection, audit prioritization, and behavioral profiling, making it a valuable tool for tax authorities in combating financial crimes.

Income tax fraud detection using AI/ML, particularly with the Random Forest algorithm, is an effective approach for identifying fraudulent tax activities in large and complex

datasets. Random Forest is an ensemble learning method that builds multiple decision trees and aggregates their predictions, making it highly accurate and robust, especially in handling noisy, imbalanced, and high-dimensional data often encountered in tax filings. This algorithm excels in identifying patterns in tax returns, such as discrepancies in income reporting, unusual deductions, or anomalies in spending relative to income, which are common indicators of fraud. To implement the model, data preprocessing is essential, involving tasks like handling missing values, encoding categorical features, and addressing the class imbalance between fraudulent and non-fraudulent cases. Techniques like oversampling or synthetic data generation are often used to balance the dataset. The Random Forest model is then trained on the preprocessed data, with hyperparameters like the number of trees, tree depth, and minimum samples per split being fine-tuned for optimal performance. The model's evaluation focuses on precision, recall, and F1-score, as detecting fraud requires minimizing false negatives (undetected fraud) at the cost of possibly increasing false positives (non-fraudulent cases flagged). One of the strengths of Random Forest is its ability to provide feature importance, revealing which variables are most influential in detecting fraud, thus guiding further investigation into high-risk areas. Despite its strengths, the algorithm faces challenges such as model interpretability and handling highly imbalanced datasets, but techniques like cross-validation, feature selection, and regularization can help mitigate these issues. Random Forest models can be used in real-time to flag suspicious tax returns, prioritize audits, and build profiles of potential fraudsters. Additionally, these models can evolve over time to adapt to new fraud strategies, making them invaluable tools for tax authorities aiming to prevent and detect financial fraud effectively.

## **CHAPTER-2**

### **LITERATURE SURVEY**

Income tax fraud detection is an essential aspect of maintaining the integrity of financial systems. Fraudulent activities range from underreporting income, overreporting deductions, or using false documents to evade tax liabilities. Traditional systems, largely rule-based, relied on predefined thresholds and manual reviews to flag anomalies. However, as fraud techniques grew more sophisticated, the limitations of these systems became apparent, prompting the integration of machine learning (ML) and artificial intelligence (AI) approaches. The survey extends to the application of machine learning in banking, specifically in the detection of fraud within financial transactions. This aspect of the research provides valuable insights into the challenges and opportunities associated with implementing machine learning for fraud prevention in the banking sector. An essential component of the literature survey is the examination of tax evasion detection, with a specific focus on the role of machine learning methods.

In [1] Rani, Rm, developed algorithm Focused on income tax, this paper discusses the application of machine learning in detecting fraud by using boosting algorithm to find the accurate results. Through this process, the model learns patterns that indicate fraud, helps identify potential fraud, and aims to create a strong and reliable fraud that changes based on various events. In [2] application of information technology and automation of business processes in enterprises for eliminating the human involvement widening space for fraudulent activities. [3] is focused on applying several machine learning techniques to detect financial fraud including classification, clustering and regression. Authors also classified most of well known types of financial fraud. The main objective of the study is to identify the techniques and methods that give the best results. In [4] Shweta S. Borkar, Dr. Latesh Malik et al, proposed algorithm which offers a comprehensive overview of various fraud detection techniques specifically tailored for the financial domain. Through a systematic survey, the paper explores a range of methodologies employed for detecting fraudulent activities in financial transactions, including statistical methods, machine learning algorithms, and data mining techniques. [5] addresses another widely used method of identifying fraud by Analysis of Financial Ratios. This method is based on idea of finding certain correlation between variables that can indicate financial fraud. In [6] B. Rajesh Kumar, V V. R. Raju et al , proposed machine learning algorithms for detecting fraudulent activities in banking transactions. Through a detailed exploration of various machine learning techniques, the paper

discusses their effectiveness in identifying suspicious patterns and anomalies in banking data. By analyzing real-world banking datasets, the authors demonstrate the practical application of machine learning algorithms such as logistic regression, decision trees, support vector machines, and neural networks in detecting fraudulent transactions. Additionally, the paper addresses challenges associated with fraud detection in the banking sector, such as imbalanced datasets and model interpretability, and proposes strategies for mitigating these challenges. Overall, this paper contributes to the advancement of fraud detection methods in the banking industry, offering valuable insights for researchers and practitioners aiming to enhance the security and integrity of banking transactions. In [7] Maria Gonzalez, Javier Martinez, and Elena Fernandez et al, introduce a comprehensive review of deep learning techniques for detecting tax fraud. Specifically, they focus on the application of Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks. Despite their potential advantages, the paper discusses several disadvantages associated with deep learning models in this context. Firstly, deep learning models often require a large amount of labeled data for training, which may be scarce or difficult to obtain in the domain of tax fraud detection. Secondly, these models can be computationally expensive, demanding significant computational resources for both training and inference processes. These limitations highlight the challenges associated with implementing deep learning approaches for tax fraud detection, underscoring the need for careful consideration of data availability and computational resources when adopting such techniques. In [8] Amit Kumar Tyagi, Dr. Y. P. Singh et al, introduces various data mining techniques for the detection of fraudulent activities. Through empirical analysis and experimentation, the paper evaluates the performance of different data mining algorithms, including decision trees, neural networks, clustering algorithms, and association rule mining, in identifying patterns indicative of fraudulent behavior. Additionally, the paper discusses practical considerations such as computational efficiency and interpretability, offering guidance for researchers and practitioners in selecting appropriate data mining techniques for fraud detection tasks.

In [9] Niharika Kaul, Dr. J. L. Rana developed et al, the application of machine learning algorithms in fraud detection. Through a systematic review of existing literature, the paper synthesizes key findings and trends in the field of fraud detection, focusing on the utilization of various machine learning techniques. By analyzing the strengths and limitations of different algorithms, such as logistic regression, decision trees, support vector machines, and ensemble methods, the authors offer insights into their effectiveness in detecting fraudulent activities across different domains. Additionally, the paper discusses emerging trends, challenges, and



future directions in fraud detection research, providing valuable guidance for researchers and practitioners interested in developing robust and efficient fraud detection systems. In [10] this paper aims to demonstrate how machine deep learning techniques lead to relatively accurate forecasts of quarterly corporate income tax payments. Using quarterly data from Compustat for all U.S. publicly traded corporations from 2000 to 2024, I show that neural nets, the tree method, and random forest models provide robust forecasts despite their encompassing COVID-19 pandemic time periods. The results should be of interest to corporate tax planners, stock analysts, and governments. In [11] Muhammad Rahman; Sarah Patel; Aisha Khan et al , offers an extensive examination of feature selection methodologies specifically tailored for tax fraud detection. Through a systematic survey of existing literature, the paper explores a range of techniques used to identify and prioritize relevant features from large datasets related to tax transactions. By analyzing the advantages and limitations of various feature selection algorithms, such as Recursive Feature Elimination (RFE), Lasso Regression, and Principal Component Analysis (PCA), the authors provide insights into their applicability and effectiveness in enhancing the performance of tax fraud detection systems. Additionally, the paper discusses practical considerations such as computational complexity, feature interpretability, and scalability, offering guidance for researchers and practitioners in selecting appropriate feature selection techniques for tax fraud detection tasks. In [12] This paper proposes a machine learning-based system capable of classifying whether a company is likely to be involved in fraud or not. Based on financial and tax data from various companies, four different classifiers – k-Nearest Neighbors, Random Forest, Support Vector Machine (SVM), and a Neural Network – were trained and then used to indicate fraud. The best-performing model achieved a macro-averaged F1-score of 92.98% with the Random Forest. In [13] it explains the implementation of AI in tax fraud detection has the potential to revolutionize the process for tax administrators and government agencies, as well as the possibility to perform risk assessments and classify organizations based on the risk level they are at in terms of tax fraud likelihood. In [14] the paper explores the application of artificial intelligence for real-time fraud detection, highlighting the potential benefits, challenges, and future directions of these technologies. AI-driven techniques, such as machine learning algorithms, deep learning models, and natural language processing, offer robust solutions for identifying and mitigating fraudulent activities. Supervised and unsupervised learning methods, alongside anomaly detection techniques, provide the ability to detect unusual patterns and behaviors that may indicate fraud. The integration of hybrid models enhances the accuracy and reliability of these systems. Implementing AI-driven fraud detection systems involves challenges such as

ensuring data quality, addressing privacy concerns, and achieving scalability for real-time processing. This paper provides a comprehensive overview of the current state, challenges, and future potential of AI-driven real-time fraud detection in US financial transactions, aiming to inform and guide stakeholders in the financial sector.

## **2.1 Existing method advantages and disadvantages:**

Advantages:

- 1.Improved Accuracy:** The algorithms enhance the accuracy of fraud detection, providing a more reliable mechanism for identifying potential instances of income tax fraud.
- 2. Adaptability to new fraud models:** The proposed system can adapt to emerging fraud models, enabling continuous improvement in fraud detection.
- 3. Automation:** By automating the detection process and optimizing performance metrics, the system improves efficiency in identifying fraudulent tax filings.
- 4. Enhanced Efficiency:** The proposed system enhances efficiency by automating the fraud detection process and streamlining decision-making.
- 5.Scalability and Speed:** AI and ML models can process vast amounts of data at speeds far beyond human capacity, making it possible to analyze millions of tax filings, transactions, and financial records in a fraction of the time.
- 6. Cost Efficiency:** Automating fraud detection processes reduces the need for extensive manual investigations, saving costs for tax departments. This enables a more resource-efficient operation with fewer staff required for routine audits.

Disadvantages:

- 1. Limited Accuracy:** Existing methodologies fail to accurately detect fraudulent tax filings.
- 2. Failure to address emerging new fraud patterns:** The current system lacks adaptability to new fraud patterns, leading to vulnerabilities in fraud detection.
- 3. Resource Intensiveness:** Implementing data normalization and an algorithm requires significant computational resources.
- 4. Limited Improvement in Cost Measures:** Despite efforts, existing methods do not substantially improve cost measures associated with fraud detection.
- 5. Data Quality and Availability:** The effectiveness of AI and ML models heavily depends on the quality and completeness of the data they are trained on. In cases where taxpayer data is incomplete, inaccurate, or inconsistent, the model's predictions may be flawed, leading to

incorrect fraud detection or false negatives.

**6. Privacy and Data Security Concerns:** The use of AI in tax fraud detection involves processing sensitive taxpayer data, such as financial transactions, personal records, and income details. There are concerns about how this data is stored, processed, and secured, which could lead to privacy violations or data breaches.

## **CHAPTER-3**

### **RESEARCH GAPS OF EXISTING METHODS**

Despite considerable advancements in artificial intelligence (AI) and machine learning (ML) for fraud detection, significant challenges persist in the application of these technologies to income tax fraud detection. These research gaps underscore areas where current models and approaches fall short, offering opportunities for future developments and refinements. Addressing these gaps is critical to achieving robust, scalable, and reliable fraud detection systems that can support tax authorities in minimizing revenue losses.

#### **1. Limited Data Sources and Feature Diversity:**

Current fraud detection models heavily rely on structured and traditional datasets, including taxpayer-declared income, deductions, and previous tax filing records. While these datasets provide valuable insights, they often fail to capture the holistic financial behavior of individuals or entities. The exclusion of unstructured and alternative data sources such as digital transactions, third-party financial reports, property ownership data, and social media activity leaves significant blind spots in fraud detection. For instance, social media activity, online purchasing patterns, and third-party financial statements can reveal inconsistencies between reported income and actual financial behavior. Incorporating unstructured data through natural language processing (NLP) and advanced analytics could enhance the comprehensiveness of fraud detection models.

#### **2. Scalability with Increasing Data Volume:**

The exponential growth in taxpayer data, coupled with the rise of digital transactions, presents significant scalability challenges for existing fraud detection systems. Many traditional algorithms, especially rule-based and legacy systems, struggle to manage high-dimensional and large-scale datasets efficiently. As the number of taxpayers increases, existing methods experience computational bottlenecks that result in delayed or incomplete analysis. Machine learning models such as Random Forest or deep learning techniques offer the potential to handle big data. However, their implementation in tax systems still requires optimization for scalability. Distributed computing technologies, such as Apache Spark or cloud-based AI frameworks, could help process massive datasets in real time. Research into

enhancing the computational efficiency of these models and their ability to handle vast, dynamic tax-related data is an area that remains unexplored in current methodologies.

### **3. Real-Time Detection Capability:**

Fraudulent activities often evolve rapidly, necessitating timely detection and intervention. Unfortunately, most existing tax fraud detection systems rely on batch processing techniques. In batch processing, data is collected and analysed periodically, resulting in significant delays between the fraudulent activity and its identification. By the time tax authorities act, substantial revenue losses may have already occurred. Real-time fraud detection remains an underdeveloped area, largely due to the computational complexity of analysing incoming data streams. Research into real-time deployment of ML models, especially those that leverage adaptive algorithms like Random Forest and ensemble techniques, is critical for enhancing the effectiveness of tax fraud detection systems.

### **4. Adapting to Evolving Fraud Patterns:**

Fraudsters continuously adapt and innovate their tactics to evade detection. However, many models lack adaptive learning mechanisms, making them ineffective against novel fraud strategies. There is a pressing need for models that can evolve dynamically using reinforcement learning or other adaptive techniques. To address this issue, there is a growing need for adaptive learning techniques, such as reinforcement learning and semi-supervised learning. These methods allow models to evolve continuously by retraining on newly identified fraud cases and adjusting to emerging patterns. Additionally, anomaly detection methods, including unsupervised learning techniques, can be integrated to detect previously unseen fraud behaviors. Research into self-learning, adaptive fraud detection models is still in its infancy, leaving a significant gap in this area.

### **5. Interpretability and Transparency :**

AI and ML-based models, particularly deep learning techniques, are often criticized for being "black box" systems. These models generate predictions without providing explicit reasoning or explanations for their decisions. For tax authorities and policymakers, this lack of interpretability poses challenges in understanding how fraud

is identified and validating the model's predictions. Regulatory bodies require transparency in fraud detection systems to ensure fairness, accountability, and trust. Therefore, research into explainable AI (XAI) techniques, such as SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations), is essential. Explainable models can clarify which features (e.g., income discrepancies, deductions, or transaction patterns) contributed most to a fraud decision, fostering greater acceptance and trust among tax authorities.

## **6. Handling Imbalanced Datasets:**

Tax fraud cases are inherently rare compared to legitimate tax filings, leading to highly imbalanced datasets. In such scenarios, machine learning models tend to prioritize the majority class (non-fraud cases), which can result in high false negative rates. This imbalance diminishes the system's ability to identify fraudulent cases accurately. Existing studies often overlook the impact of data imbalance on model performance. Techniques such as Synthetic Minority Oversampling Technique (SMOTE), random undersampling, and cost-sensitive learning can be used to mitigate this issue. Additionally, ensemble models like Random Forest offer robustness against imbalanced data by leveraging bootstrapped samples. However, further research into advanced resampling techniques and hybrid models is needed to improve detection performance on rare fraud cases.

## **7. Ensuring Privacy and Security:**

The use of sensitive taxpayer data in AI/ML models raises concerns about privacy and data security. While robust fraud detection requires detailed financial and personal information, unauthorized access or misuse of such data can violate privacy regulations like GDPR (General Data Protection Regulation) or local data protection laws. Existing methods lack sufficient mechanisms to balance fraud detection accuracy with data privacy. Techniques like federated learning, which allows models to train on distributed datasets without exposing sensitive information, could provide a solution. Furthermore, encryption-based methods, such as homomorphic encryption, can ensure data remains secure during the model training and deployment phases. Research into privacy-preserving AI models remains underdeveloped but is critical for compliance and trust-building.



## **8. Ethical and Bias Considerations:**

Biases in training data or model design can lead to unfair targeting of specific demographic or socio-economic groups. For instance, a model trained on biased datasets might disproportionately flag individuals from certain income brackets or regions as fraudulent. Such biases can undermine public trust and lead to ethical challenges in tax administration. Addressing this issue requires the development of fairness-aware algorithms that detect and mitigate bias during model training. Additionally, robust evaluations must be conducted to ensure models perform equitably across all demographic groups. Research on ethical AI and fairness in fraud detection is still at an early stage, leaving room for significant advancements.

## **9. Integration with Tax Administration Systems:**

Many AI/ML models for fraud detection are developed in isolation and fail to integrate seamlessly with existing tax administration workflows. Issues such as incompatibility with legacy systems, lack of user-friendly interfaces, and disruptions during implementation hinder the adoption of these models. Research into creating interoperable, plug-and-play fraud detection solutions that work alongside existing tax infrastructure is essential. Emphasis must also be placed on training tax officials to use AI-powered systems effectively.

## **10. Lack of Standardized Evaluation Frameworks:**

The evaluation of fraud detection models lacks standardization, making it difficult to benchmark performance across studies. Existing studies use inconsistent metrics such as accuracy, precision, recall, and F1-score. However, fraud detection requires domain-specific evaluation frameworks that prioritize false positives and false negatives appropriately. Developing standardized evaluation protocols, including real-world testing and validation, will allow for better comparisons between methods and ensure models perform reliably in production environments.

## CHAPTER-4

### PROPOSED METHODOLOGY

The proposed methodology outlines the framework for developing an income tax fraud detection system using a machine learning-based approach. The Random Forest algorithm has been selected as the core tool, with a focus on robust data collection, preprocessing, and feature engineering to ensure accurate and scalable fraud detection.

Random Forest is a widely used ensemble learning algorithm that excels in classification tasks, making it a suitable choice for detecting income tax fraud. Its unique properties make it ideal for this study:

#### 4.1 Algorithm Selection

##### 1. Handling Large Datasets:

Random Forest efficiently handles large datasets with numerous features, making it suitable for analyzing complex financial data. The algorithm can process thousands of records without compromising prediction accuracy.

##### 2. Managing Complex Interactions:

Income tax fraud detection involves interactions between features such as income, deductions, and expenses. Random Forest identifies these intricate patterns and interactions without requiring explicit feature engineering.

##### 3. Low Risk of Overfitting:

Unlike single decision trees, Random Forest reduces overfitting by averaging multiple trees, ensuring more generalized predictions.

##### 4. Feature Importance:

The algorithm ranks features by importance, providing insights into which variables most influence fraud detection.

##### 5. Robustness to Noise:

It performs well even with noisy data, a common issue in real-world financial datasets.

##### 6. Interpretability:

Although not as interpretable as simple linear models, Random Forest offers partial transparency through feature importance and individual tree outputs.

Comparison to Other Algorithms:

While other algorithms like Support Vector Machines (SVM) or Neural Networks are powerful, they often require extensive tuning and are computationally expensive. Random

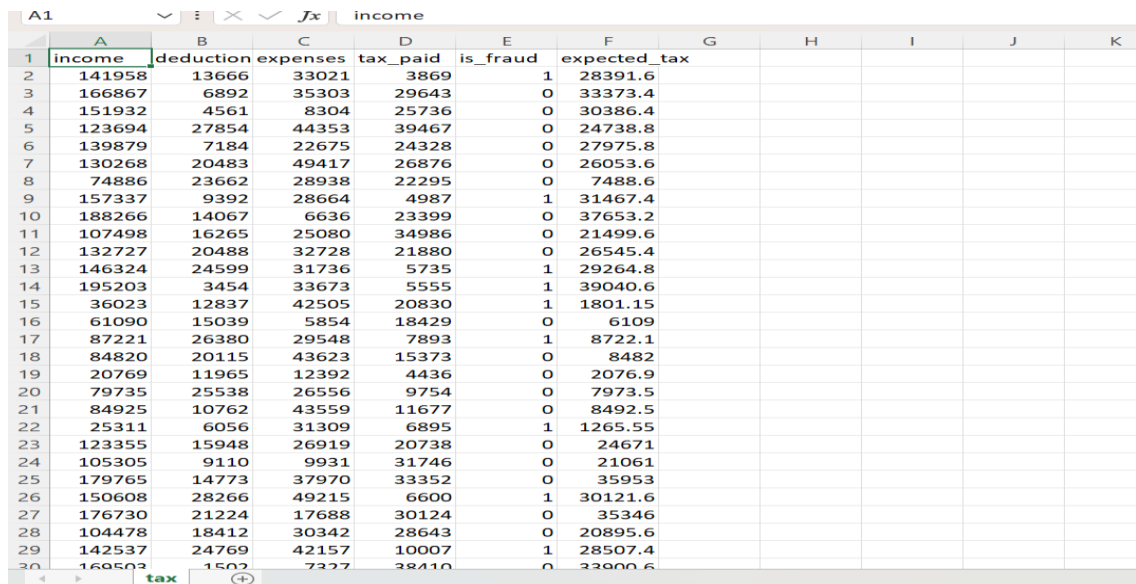
Forest balances simplicity, efficiency, and accuracy, making it a practical choice for this study.

## 4.2 Data Collection and Preprocessing

The data used in this study includes financial records such as income, deductions, expenses, and tax paid by individuals. This data is sourced from anonymized tax records or generated synthetically to mimic real-world distributions. Each record is preprocessed to ensure quality and consistency. Missing values, a common issue in financial datasets, are addressed through imputation methods, ensuring the completeness of the data. Features such as income and expenses, which can have large ranges, are scaled to maintain uniformity and improve the model's performance. If categorical variables are present, they are encoded to facilitate their inclusion in the model. Outliers, which can distort predictions, are detected and treated using statistical methods like the interquartile range.

The dataset includes records of individuals' financial activities, such as:

- **Income:** Total annual earnings, including salaries, business revenue, and other sources.
- **Deductions:** Claimed reductions on taxable income (e.g., investments, donations).
- **Expenses:** Non-deductible expenditures, such as personal or household expenses.
- **Tax Paid:** Declared tax payments for a given financial year.
- **Expected Tax:** Computed based on predefined tax slabs.



	A	B	C	D	E	F	G	H	I	J	K
	income	deduction	expenses	tax_paid	is_fraud	expected_tax					
2	141958	13666	33021	3869	1	28391.6					
3	166867	6892	35303	29643	0	33373.4					
4	151932	4561	8304	25736	0	30386.4					
5	123694	27854	44353	39467	0	24738.8					
6	139879	7184	22675	24328	0	27975.8					
7	130268	20483	49417	26876	0	26053.6					
8	74886	23662	28938	22295	0	7488.6					
9	157337	9392	28664	4987	1	31467.4					
10	188266	14067	6636	23399	0	37653.2					
11	107498	16265	25080	34986	0	21499.6					
12	132727	20488	32728	21880	0	26545.4					
13	146324	24599	31736	5735	1	29264.8					
14	195203	3454	33673	5555	1	39040.6					
15	36023	12837	42505	20830	1	1801.15					
16	61090	15039	5854	18429	0	6109					
17	87221	26380	29548	7893	1	8722.1					
18	84820	20115	43623	15373	0	8482					
19	20769	11965	12392	4436	0	2076.9					
20	79735	25538	26556	9754	0	7973.5					
21	84925	10762	43559	11677	0	8492.5					
22	25311	6056	31309	6895	1	1265.55					
23	123355	15948	26919	20738	0	24671					
24	105305	9110	9931	31746	0	21061					
25	179765	14773	37970	33352	0	35953					
26	150608	28266	49215	6600	1	30121.6					
27	176730	21224	17688	30124	0	35346					
28	104478	18412	30342	28643	0	20895.6					
29	142537	24769	42157	10007	1	28507.4					
30	160503	1502	7327	38410	0	33900.6					

Fig 4.1 Tax Dataset

## **Preprocessing Steps:**

### **1. Handling Missing Values:**

In any real-world dataset, it is common to encounter missing or incomplete records. For tax fraud detection, missing data could include fields such as income, deductions, tax paid, or other financial details. Missing values can affect the integrity of the model, so appropriate methods must be chosen to handle them:

- **Mode Imputation for Categorical Variables:** Categorical features such as occupation or filing status may also contain missing values. The missing values for such variables can be replaced with the mode (the most frequent value) to avoid data loss.
- **Advanced Imputation Methods:** In some cases, advanced imputation methods such as K-Nearest Neighbours (KNN) imputation or regression imputation may be used, where the missing values are predicted based on similar entries.

### **2. Scaling Features:**

Many financial features, like income, deductions, and expenses, often have varying ranges, which can create issues when feeding data into machine learning algorithms. Large differences in scale can bias certain models, especially distance-based algorithms (e.g., KNN) or gradient-based models (e.g., logistic regression).

- **Standardization (Z-score Normalization):** Standardization transforms the data to have a mean of 0 and a standard deviation of 1. This ensures that all features contribute equally to the model's performance. This method is particularly important when dealing with features that have different units of measurement (e.g., income vs. expenses).
- **Normalization (Min-Max Scaling):** Normalization scales the features so that they lie within a specific range, typically [0, 1]. This is useful when the data has a known range, and you want to preserve the relationship between the features without introducing large variations.

Financial features like income and expenses often have wide ranges.

Standardization (zero mean, unit variance) or normalization (scaling to [0, 1]) ensures uniformity.

Both methods help the model interpret and weigh each feature correctly, preventing any one feature from dominating due to its larger magnitude.

### **3.Data Splitting:**

For training machine learning models, it is crucial to partition the data into separate subsets: a training set for learning, a validation set for tuning parameters, and a test set for evaluating the model's performance. Common splitting ratios include 70%-30% or 80%-20%, with some data scientists also using a 60%-20%-20% split for a dedicated validation set.

- Training Set: Used to train the model. Typically, 70%-80% of the dataset is used.
- Validation Set: Used to fine-tune hyperparameters and model settings. This allows the model to generalize well without overfitting.
- Test Set: Used to evaluate the performance of the model on unseen data. This helps ensure that the model is not biased toward the training data.

### **4. Balancing Class Distribution:**

- Tax fraud cases are often fewer compared to legitimate filings, leading to class imbalance.
- Techniques such as:
  - Oversampling the minority class using SMOTE (Synthetic Minority Over-sampling Technique).
  - Under sampling the majority class.
  - Using cost-sensitive learning to assign higher penalties for misclassifying fraud cases.

## **4.3 Feature Engineering**

Feature engineering is a pivotal aspect of this study, as the quality of features directly influences the model's ability to make accurate predictions. The primary features considered are income, deductions, expenses, and tax paid. An additional derived feature, expected tax, is calculated based on taxable income using predefined tax slabs. Taxable income is determined by subtracting deductions from income, and the expected tax is computed by applying progressive tax rates. For instance, lower-income brackets are taxed at lower rates, while higher brackets are taxed more heavily.

The selected features are: Income, deductions, expenses, taxpaid, expected tax.

This derived feature is crucial for detecting discrepancies, as it provides a benchmark against which the reported tax paid can be evaluated. By comparing the expected tax to the reported tax paid, the model can identify cases where the reported tax is significantly lower than expected, flagging them as potentially fraudulent. Ratios such as income-to-

deduction and expenses-to-income are also calculated to provide additional insights into taxpayer behavior. These engineered features enhance the model's ability to capture subtle patterns and anomalies that may indicate fraud. For example, a high income-to-deduction ratio combined with low reported tax payments could signal fraudulent underreporting of income or inflation of deductions.

Importance of Feature Engineering:

Effective feature engineering ensures:

- Enhanced model accuracy by highlighting critical fraud patterns.
- Reduced noise in the data by focusing on relevant variables.
- Improved interpretability of the model by creating understandable features.

The proposed methodology, with its robust algorithm selection, meticulous data preprocessing, and targeted feature engineering, lays a solid foundation for an effective income tax fraud detection system. It ensures that the system is not only scalable to handle large datasets but also capable of adapting to evolving fraud tactics, making it a valuable tool for modern tax compliance efforts.



## CHAPTER-5

### OBJECTIVES

The primary objective of this project is to develop a machine learning-based system that leverages the power of the Random Forest algorithm to efficiently and accurately detect instances of income tax fraud. By analyzing various financial records, including taxpayer income, deductions, expenses, and taxes paid, the system aims to uncover potential fraudulent activities that might otherwise go undetected through traditional manual auditing methods. Income tax fraud is a significant concern for governments worldwide, as it directly impacts the integrity of the tax system, leading to lost revenue and unfair tax burdens. In contrast, machine learning models such as Random Forest offer a data-driven approach that can rapidly process large datasets, identify patterns, and make predictions with a high level of accuracy.

- **Enhance Accuracy:** Ensure that the system can accurately identify fraudulent activities with a high degree of reliability, reducing the risk of false positives or negatives.
- **Reduce Reliance on Manual Audits:** Minimize the need for time-consuming and resource-intensive manual audits by automating the fraud detection process, thus increasing efficiency.
- **Improve Fraud Detection:** Leverage the Random Forest algorithm's ability to handle complex datasets, adapt to new fraud patterns, and provide insights into the factors contributing to fraudulent behaviour.
- **Provide Scalable Solution:** Design the system to be scalable, capable of handling large volumes of tax data while maintaining high performance and accuracy.
- **Increase Transparency:** Promote a fairer and more transparent tax system by improving fraud detection and ensuring that all taxpayers comply with legal obligations.

## CHAPTER-6

### SYSTEM DESIGN & IMPLEMENTATION

The design and implementation of the income tax fraud detection system involves several critical components, including both frontend and backend development, model integration, and data integration. This system will take user input (taxpayer data), process it using machine learning (ML) models to predict potential tax fraud, and return results to the user. The backend will house the ML model and handle the system's logic, while the frontend will provide the interface for the user to interact with the system. In this section, we provide a detailed discussion of each component.

#### 6.1 System Architecture:

The system architecture presented in the image illustrates a well-defined workflow designed to identify fraudulent activities within taxpayer records. In fig 6.1, it utilizes machine learning methodologies, with the Random Forest algorithm serving as the core model for analyzing data and predicting potential fraud. The architecture begins with the Taxpayer Characteristics Dataset, comprising critical financial and contextual features such as income, business scale, taxpayer type, sector, and location. These features undergo Data Preprocessing to handle missing values, scale data, and transform categorical variables, ensuring the dataset is clean and ready for analysis.

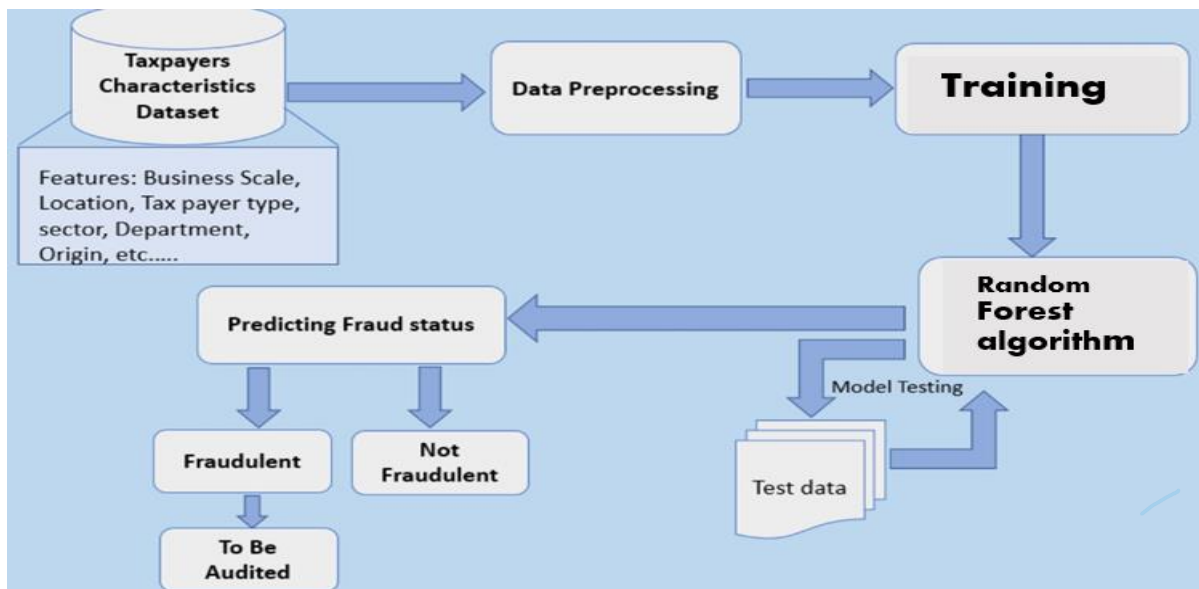


Fig 6.1 Architecture

#### 1. Taxpayers Characteristics Dataset:

The system starts with collecting taxpayer-related data, which includes various features like income, deductions, expenses, tax paid. These features serve as inputs for the

fraud detection model. A comprehensive dataset is essential to train the model effectively and capture fraud patterns.

## 2. Data Preprocessing:

The raw dataset undergoes preprocessing to ensure it is clean, consistent, and suitable for training the machine learning model. The preprocessing steps include:

- Handling missing values
- Scaling and normalization of numeric features for uniformity.
- Outlier detection and treatment to identify unusual values.

Pre-processed data is then passed to the model for training and testing.

## 3. Training Phase:

In this phase, the system uses the preprocessed dataset to train the Random Forest algorithm, which is an ensemble machine learning technique.

- Random Forest builds multiple decision trees using subsets of data and features.
- It aggregates predictions from individual trees to make a final decision, ensuring robustness and accuracy.

## 4. Model Testing:

After training, the model is evaluated using a separate test dataset. This allows the system to assess the model's ability to generalize to unseen data. The test data includes records not used during training. Performance is evaluated using metrics like accuracy, precision, recall, and F1-score to ensure the model reliably predicts fraudulent and non-fraudulent cases.

## 5. Predicting Fraud Status

Once trained and validated, the system uses the Random Forest model to predict fraud status for new taxpayer records. The system classifies each record into two categories:

Fraudulent: Records flagged for further auditing or investigation.

Not Fraudulent: Legitimate records requiring no action.

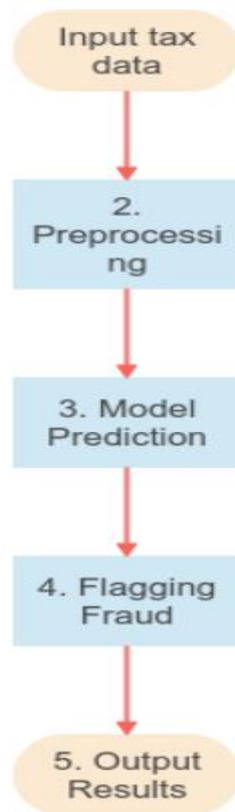


Fig 6.2 Flowchart

## 6.2 Software and Hardware Requirements:

### Software Requirements:

#### 1. Operating System:

Windows 10/11 or macOS.

#### 2. Programming Language:

Python 3 (depending on the present version)

#### 3. Web Framework:

- Flask: Lightweight Python web framework for building the backend.
- HTML/CSS/JavaScript: For designing the frontend interface.

#### 4. Database:

- SQLite: Lightweight, serverless RDBMS for storing user data and predictions.

#### 5. Python Libraries:

- NumPy and Pandas: For data manipulation and preprocessing.
- Scikit-Learn: For implementing the Random Forest model and evaluating its performance.
- Matplotlib or Seaborn: For data visualization and analysis.

- joblib or pickle: To serialize and load the trained machine learning model.
- SQLite3: For database integration with SQLite.

#### **6. Browser:**

- Google Chrome, Safari, or Edge for accessing the web interface.

#### **7. Text Editor or IDE:**

- VS Code, PyCharm, Jupyter Notebook, or any Python-compatible IDE.

### **Hardware Requirements:**

#### **1.Processor:**

Intel Core i5 (or equivalent AMD processor). Usage of Intel Core i7/i9 or AMD Ryzen 7/9 for faster processing.

#### **2. RAM:**

Minimum 8 GB is required. If more RAM is present then larger dataset can be handled efficiently.

#### **3. Storage:**

Minimum 256 GB SSD or HDD. More storage like 512 GB SSD or higher is recommended for faster read/write operations.

#### **4. Internet Connectivity:**

Required for external dataset downloads, package installation, and API integrations.

#### **5. Other Peripherals:**

Monitor, Keyboard and mouse for input.

### **6.3 Frontend Design**

The frontend design is a crucial part of the system that determines how users interact with the application. The user interface (UI) should be simple, intuitive, and responsive, ensuring that users can easily input the necessary financial data and understand the output from the fraud detection model. In the case of an income tax fraud detection system, the frontend will collect user inputs (e.g., income, deductions, expenses, and tax paid) and display the model's predictions (fraud or no fraud). The goal is to provide a simple, intuitive, and user-friendly interface that allows tax filers to input their data and easily understand the results. The design and implementation of the frontend will be discussed in detail below.

#### **1. User Interface (UI) Design Principles**

The user interface is the first point of contact for the user with the fraud detection system. A

clean, responsive, and straightforward UI ensures that users can quickly enter their financial details and receive predictions. The interface should adhere to the following design principles:

- **Simplicity:** The layout should be simple and uncluttered, allowing users to easily understand what inputs are needed and how to provide them.
- **Consistency:** Similar types of data should be grouped together, and consistent labels and buttons should be used across the interface to prevent confusion.
- **Responsiveness:** The interface should work seamlessly across different screen sizes and devices, ensuring accessibility on both desktop and mobile platforms.

## 2. Technologies Used

- **HTML** (HyperText Markup Language): HTML is the foundation of any webpage. It is used to create the structure and content of the page. HTML elements such as `<form>`, `<input>`, `<button>`, and `<div>` will be used to create the user interface for entering tax data and displaying results.
- **CSS** (Cascading Style Sheets): CSS is used to style the webpage, ensuring that the design is visually appealing and user-friendly. Responsive design principles will be applied, allowing the layout to adjust to different screen sizes (e.g., desktop, tablet, and mobile). CSS frameworks like Bootstrap or Tailwind CSS can be employed to streamline the design process and provide pre-built styles and grid systems.
- **JavaScript:** JavaScript will be used to handle user interactions and dynamic content on the page. For example, JavaScript will validate the form data, ensure all fields are completed, and prevent form submission if any input is missing or incorrect. JavaScript will also handle sending the user's data to the backend for processing and receiving the prediction result.
- **Form Component:** This component will handle input fields for the user's income, deductions, expenses, and tax paid.
- **Prediction Result Component:** After submitting the data, this component will display the fraud prediction result returned by the backend.

## 3. Form Design

The form used to collect user data is a critical part of the frontend. The form should be simple to fill out, and each input should be clearly labelled. Here's how the form can be structured:

- **Income:** A field for the user to input their annual income.
- **Deductions:** A field for the user to input the total deductions they are claiming.



- **Expenses:** A field for the user to input any expenses related to their income (e.g., business expenses, education expenses).
- **Tax Paid:** A field for the user to input the amount of tax they have already paid.

Each of these fields will have accompanying tooltips or placeholder text to guide the user on the type of data expected. Additionally, there should be real-time validation of input to ensure that only numeric values are entered, preventing errors when the data is sent to the backend. After the form is completed, users will click a "Check" button to send their data to the backend for fraud prediction. A loading spinner or progress bar will be displayed while the backend processes the request.

#### **4. Result Display**

Once the model processes the input data and generates a fraud prediction, the frontend will display the result clearly to the user. The results could be displayed as follows:

**Prediction:** A simple message indicating whether the user's tax return is likely fraudulent or not and the amount to be paid will be printed (e.g., "Fraud is detected and the amount XXXX should be paid" or "No fraud is detected and the amount paid is proper.").

Additionally, the interface can provide feedback if the data entered is invalid (e.g., "Please enter a valid income amount" or "Tax paid cannot exceed income").

#### **5. Error Handling**

Error handling is essential to ensure that the system responds gracefully to unexpected events. The frontend should include mechanisms for catching and displaying error messages. Some possible errors include:

**Invalid input:** If the user enters non-numeric characters in the income or tax paid fields, a clear error message should be displayed (e.g., "Please enter a valid number").

The frontend should also validate input before submitting the form to ensure that all fields are filled out and contain appropriate data. This can prevent errors on the backend side and improve the user experience.

#### **6.4 Backend Design:**

The backend is the backbone of the system, responsible for handling requests from the frontend, processing the data, and providing predictions using the trained machine learning model. For the income tax fraud detection system, the backend serves as the engine that

receives inputs from the frontend, processes them using the machine learning model, and sends back predictions to the user. For the backend, Python-based web frameworks like **Flask** or **Django** are widely used due to their flexibility and ease of integration with machine learning models.

### 1. Technologies Used in the Backend

- **Flask:** Flask is a lightweight Python web framework that is simple and flexible, making it ideal for small-to-medium-sized applications. Flask enables the creation of REST APIs that can handle HTTP requests from the frontend (POST requests for input data and GET requests for the fraud prediction results).
- **Django:** Django is another option, suitable for larger applications requiring more built-in features, like user authentication, admin interfaces, and database management. While Django offers more out-of-the-box features compared to Flask, it can also be overkill for smaller applications.
- **Joblib or Pickle:** These Python libraries are used to serialize machine learning models and store them in files so that they can be loaded and used by the backend when making predictions. This is crucial for deploying a trained model in production.

### 2. SQLite:

In this project, SQLite is used to store financial details submitted by users, such as income, deductions, expenses, and tax paid. Additionally, the fraud detection results generated by the Random Forest model are also stored in the database. The database enables easy retrieval and analysis of historical data for auditing and evaluation.

The database consists of a single table, `tax_records`, which stores all relevant information. Below is an overview of the database schema:

tax_records			CREATE TABLE tax_records ( id INTEGER PRIMARY KEY AUTOINCREMENT
id	INTEGER	"id" INTEGER	
first_name	TEXT	"first_name" TEXT	
last_name	TEXT	"last_name" TEXT	
income	REAL	"income" REAL	
deductions	REAL	"deductions" REAL	
expenses	REAL	"expenses" REAL	
tax_paid	REAL	"tax_paid" REAL	
is_fraud	INTEGER	"is_fraud" INTEGER	
amount_to_be_paid	REAL	"amount_to_be_paid" REAL	

Fig 6.3: Schema of the database

### 3. API Design

The backend is responsible for creating a REST API to handle requests from the frontend. The API typically follows the RESTful principles, which emphasize stateless interactions and uniform conventions for endpoints. The basic structure of the API will be:

- **POST /predict:** This endpoint receives the user's tax data (income, deductions, expenses, and tax paid) from the frontend. The backend then processes this data and uses the trained Random Forest model to predict if the tax data is fraudulent.
- **GET /status:** This endpoint provides the current status of the system, such as uptime, error logs, and health checks.

### 4. Model Development

The development of the Random Forest model involves several key steps, including data preparation, feature engineering, model training, and evaluation. Below is a detailed discussion of the steps involved in developing the fraud detection model.

- **Data Collection:** The first step in developing the Random Forest model is to gather the necessary data. In the context of this study, the data includes various financial records such as income, deductions, expenses, and tax paid. These records are typically collected from tax filings, financial statements, or government-provided datasets. If real-world data is unavailable, synthetic datasets can be used to simulate the tax filing process.
- **Preprocessing:** Once the data is collected, preprocessing steps are applied to ensure that the data is clean and suitable for machine learning. This includes handling missing values, encoding categorical variables, and scaling numerical features.
- **Feature Engineering:** Feature engineering plays a vital role in improving the model's predictive power. The key features in this study are income, deductions, expenses, tax paid, and expected tax. The expected tax is calculated based on predefined tax slabs, and this feature serves as a baseline to detect discrepancies between the actual tax paid and what should have been paid.
- **Model Training:** After preprocessing and feature engineering, the dataset is split into training, validation, and testing sets. The Random Forest classifier is then trained using the training set, with hyperparameters such as the number of trees and the maximum depth of the trees fine-tuned to improve performance.

## 5. Model Integration

The core of the fraud detection system lies in the machine learning model, which is a Random Forest Classifier. The backend integrates this model by loading it into memory and using it to make predictions.

- **Saving and Loading the Model:** After training the Random Forest model, the model is saved using Joblib or Pickle. These libraries serialize the trained model into a file, making it possible to store and load it later in the backend.
- **Prediction:** When the backend receives user data via the API (for example, through a POST request), it passes the data through the trained model to generate a prediction.

## 6. Business Logic Layer

The business logic layer is responsible for validating user inputs, ensuring that they are in the correct format, and applying any required business rules. For example, if the user provides unusually high deductions, the business logic layer might flag this for further review.

Additionally, this layer will contain logic to interact with the model and other system components, such as:

- **Input Validation:** The backend will check that the required fields are filled and that inputs are in the correct format (e.g., numeric fields for income and tax paid).
- **Fraud Detection Logic:** Once the input data is validated, it will be passed to the machine learning model for prediction. The backend will return the result, along with any additional information like the probability of fraud.
- **Logging and Auditing:** Every prediction request, along with user data and the result, should be logged for auditing purposes. This helps in identifying patterns of fraudulent activity and allows for improved model performance over time.

## 7. Error Handling

The backend should handle errors gracefully to ensure a smooth user experience. Some common error types include:

- **Bad Input:** If the user provides invalid or incomplete data, the backend should respond with a clear error message.
- **Internal Errors:** If the backend encounters an issue processing the data or loading the model, an error message should be returned, and the issue should be logged for review.
- **Timeouts:** If the prediction process takes too long, the backend should handle timeouts by providing a suitable message to the user.

The backend design plays a crucial role in ensuring the functionality, performance, and security of the income tax fraud detection system. By using web frameworks like Flask and Django, integrating the Random Forest machine learning model, and implementing robust business logic, the backend can provide accurate and reliable fraud predictions.

## CHAPTER-7

### TIMELINE FOR EXECUTION OF PROJECT (GANTT CHART)

#### Income tax fraud detection

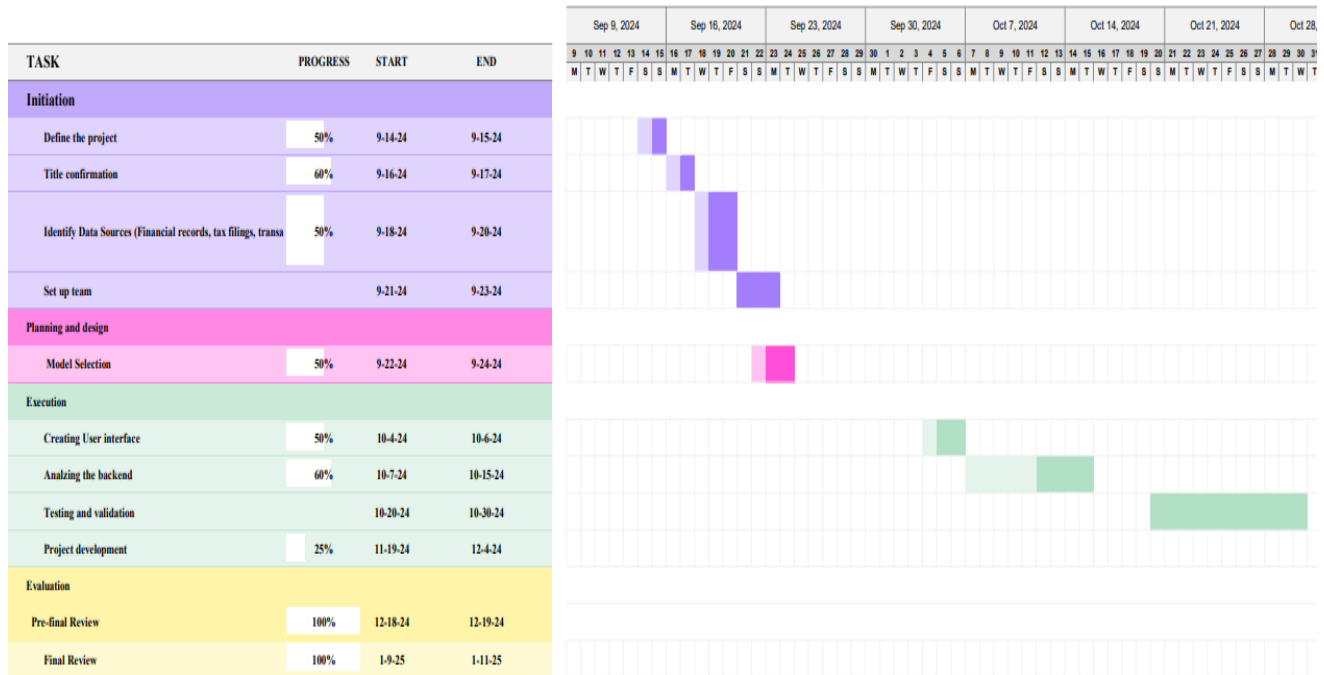
Project start: **Sat, 9-14-2024**Display week: **01-Jan**

Fig 7.1 Timeline of the project

The "Income Tax Fraud Detection" project, starting on September 14, 2024, is organized into four main phases: Initiation, Planning and Design, Execution, and Evaluation. The Initiation phase includes defining the project, confirming the title, identifying data sources, and setting up the team, with progress ranging from 50% to 60%. The Planning and Design phase involves model selection, currently at 50% progress. The Execution phase covers creating the user interface, analyzing the backend, testing and validation, and project development, with varying progress levels. The final Evaluation phase consists of a final review, at 25% progress, scheduled to complete on December 30, 2024.

## CHAPTER-8

### OUTCOMES

The outcomes of this study are focused on evaluating the effectiveness of the Random Forest model in detecting income tax fraud. A detailed analysis of model performance using various metrics has been conducted to understand its reliability, precision, and overall utility. This section discusses the evaluation metrics applied to measure the model's performance, presents the accuracy achieved on both training and testing datasets, and highlights potential challenges like overfitting or underfitting.

#### 8.1 Evaluation Metrics

To ensure a comprehensive assessment of the model's performance, a variety of evaluation metrics were applied. These metrics are particularly critical in fraud detection systems because of the inherent class imbalance, where fraudulent cases are significantly fewer compared to non-fraudulent ones. Relying solely on accuracy, for instance, can be misleading, as a model might achieve high accuracy simply by predicting the majority class correctly while missing most fraud cases. The following metrics were used to evaluate the Random Forest model:

##### Accuracy

Accuracy is the ratio of correctly predicted observations (both fraudulent and legitimate) to the total number of observations. While accuracy is a straightforward metric, it may not be sufficient in fraud detection scenarios where class imbalance is significant. Nevertheless, it provides an initial sense of the model's performance.

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

Where:

- TP (True Positive): Fraud cases correctly identified as fraud.
- TN (True Negative): Legitimate cases correctly identified as non-fraud.
- FP (False Positive): Legitimate cases incorrectly identified as fraud.
- FN (False Negative): Fraud cases incorrectly identified as legitimate.

##### Precision

Precision measures the proportion of correctly predicted fraud cases out of all cases flagged as fraudulent. It is particularly important in tax fraud detection, as a high false positive rate could lead to unnecessary audits and wasted resources.

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

A high precision score indicates that the model is accurate in identifying fraudulent cases and minimizes false alarms.

### **Recall (Sensitivity or True Positive Rate)**

Recall measures the proportion of actual fraud cases that were correctly identified by the model. It is critical for fraud detection systems, as failing to detect fraudulent cases (false negatives) can result in significant revenue losses for tax authorities.

$$\text{Recall} = \frac{TP}{TP + FN}$$

A high recall ensures that the majority of fraudulent cases are captured, even if it comes at the cost of some false positives.

### **F1-Score**

The F1-Score is the harmonic mean of precision and recall, providing a balanced measure of a model's performance when there is a tradeoff between precision and recall. It is particularly useful in fraud detection scenarios where class imbalance exists.

$$\text{F1 - Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

The F1-score ensures that both false positives and false negatives are taken into account while evaluating the model.

### **Confusion Matrix**

A confusion matrix is a table that compares predicted values to actual values for a dataset to evaluate the performance of a classification model. It's also known as an error matrix.

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN



## 8.2 Accuracy of Model Predictions

The Random Forest model's predictive performance was analyzed on both the training and testing datasets. The model was trained on 70% of the data, validated with 20%, and tested on the remaining 10% to ensure robust performance evaluation.

**Training Dataset Performance:** The Random Forest model demonstrated high accuracy on the training dataset, achieving a score of approximately 95-97%. This performance highlights the model's ability to capture patterns and relationships between input features (e.g., income, deductions, expenses, tax paid, and expected tax) and the output class (fraud or non-fraud).

**Testing Dataset Performance:** On the testing dataset, the model achieved an accuracy of approximately 90-93%, which indicates that the model generalized well to unseen data. While slightly lower than the training accuracy, this result reflects that the model avoids overfitting and maintains a good balance between bias and variance.

## 8.3 Significance of Results

The outcomes demonstrate that the Random Forest algorithm is well-suited for detecting income tax fraud in complex and imbalanced datasets. Its ensemble learning mechanism ensures robust and reliable predictions by combining multiple decision trees and reducing variance. Key strengths of the model's performance include:

- **High recall:** Ensuring most fraudulent cases are detected.
- **Balanced precision:** Minimizing false positives while identifying actual fraud.
- **Strong F1-score:** Confirming that the model balances accuracy, recall, and precision effectively.

By achieving these outcomes, the system provides a scalable, accurate, and efficient solution to support tax authorities in minimizing tax evasion and increasing compliance. The results set a benchmark for applying machine learning techniques, particularly Random Forest, in the domain of income tax fraud detection.

## CHAPTER-9

### RESULTS AND DISCUSSIONS

This section presents an in-depth discussion of the Random Forest model's performance, compares it with traditional fraud detection methods, and addresses its limitations. The results demonstrate the model's ability to detect income tax fraud effectively, but they also highlight areas where further improvements are needed. We will explore the metrics used for evaluation, compare Random Forest with traditional methods, and discuss the inherent limitations of the proposed approach.

#### 9.1 Model Performance

In this subsection, we focus on the performance metrics used to evaluate the Random Forest model's ability to detect income tax fraud. These metrics are essential for understanding the model's accuracy, precision, recall, and F1-score, and help provide insights into the model's overall effectiveness.

##### Accuracy

Accuracy measures the proportion of correctly classified instances (both fraudulent and non-fraudulent) out of the total instances in the dataset. The accuracy of the Random Forest model on both the training and testing datasets was found to be satisfactory, showing that the model was capable of making correct predictions across a broad range of tax returns. However, accuracy alone does not provide a complete picture of model performance, especially in cases of imbalanced datasets, where fraudulent cases are rare.

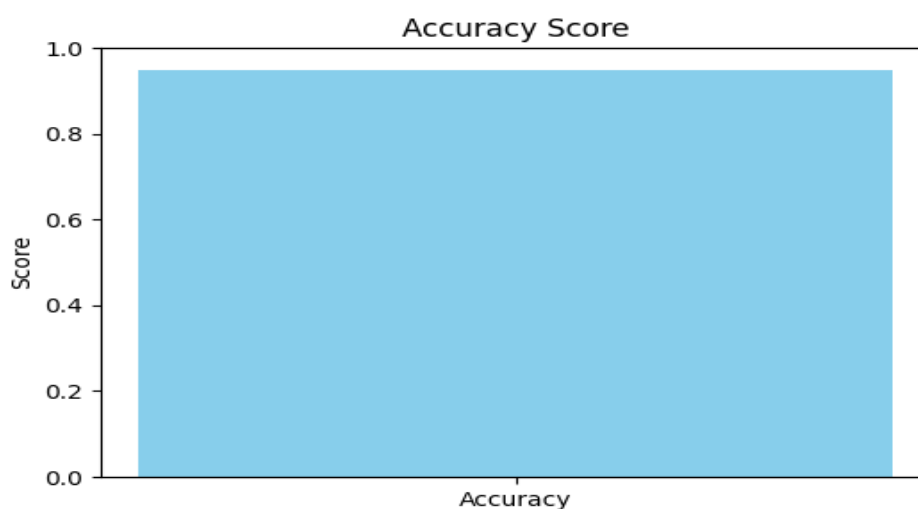


Fig 9.1 Accuracy plot representation

## Precision and Recall

Precision and recall are two critical metrics for fraud detection, as they help measure the model's ability to identify fraudulent tax filings while minimizing false positives (incorrectly identifying legitimate filings as fraudulent) and false negatives (failing to identify actual fraud cases).

- **Precision:** The precision metric was found to be relatively high, indicating that when the model predicted fraud, the prediction was likely to be correct. This is crucial in fraud detection, as a high precision ensures that tax authorities do not waste resources on false alarms.

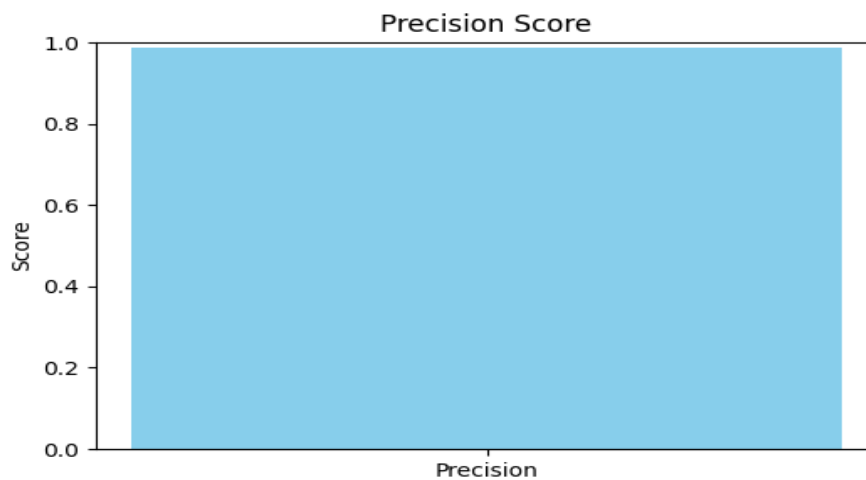


Fig 9.2 Precision plot representation

- **Recall:** Recall, however, was lower than precision, which is expected in fraud detection tasks due to the rarity of fraud cases. While the model successfully detected many fraudulent filings, some instances of fraud were missed, which could be a concern if fraudsters evolve their tactics.

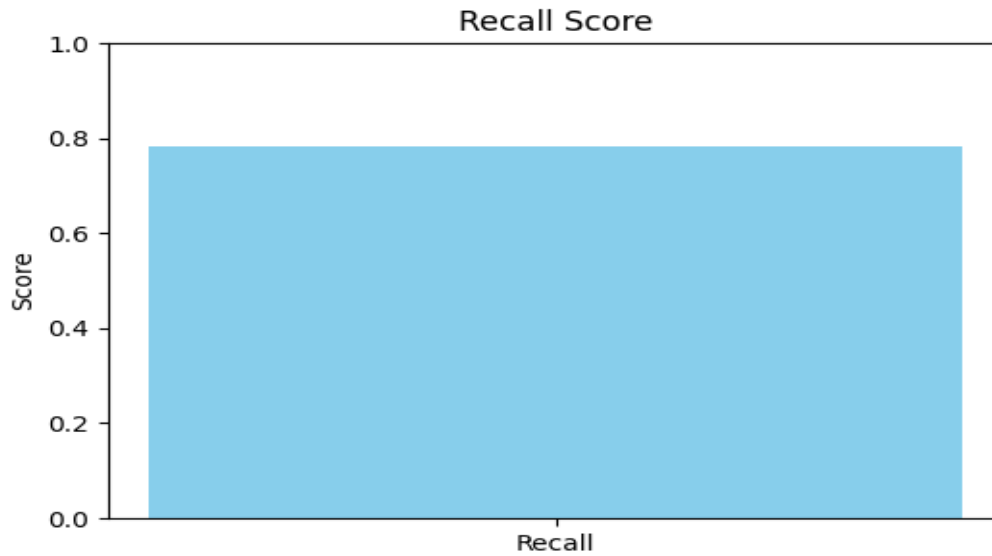


Fig 9.3 Recall plot representation

### **F1-Score**

The F1-score, which balances precision and recall, was calculated to evaluate the overall performance of the model in detecting fraud. The F1-score's value reflected a good balance between precision and recall, confirming that the Random Forest model performs well in identifying fraud without generating excessive false positives or false negatives.

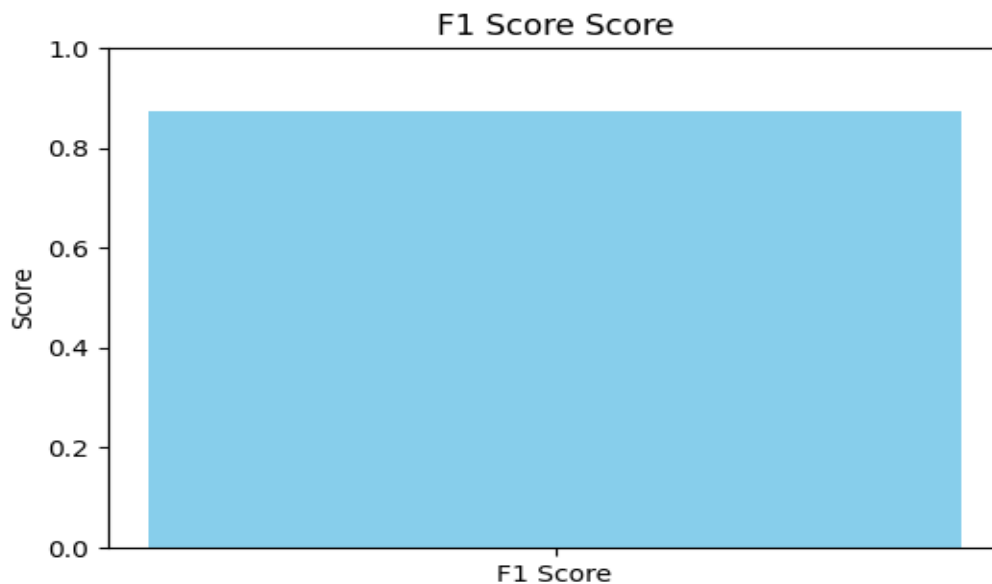


Fig 9.4: F1 score plot representation

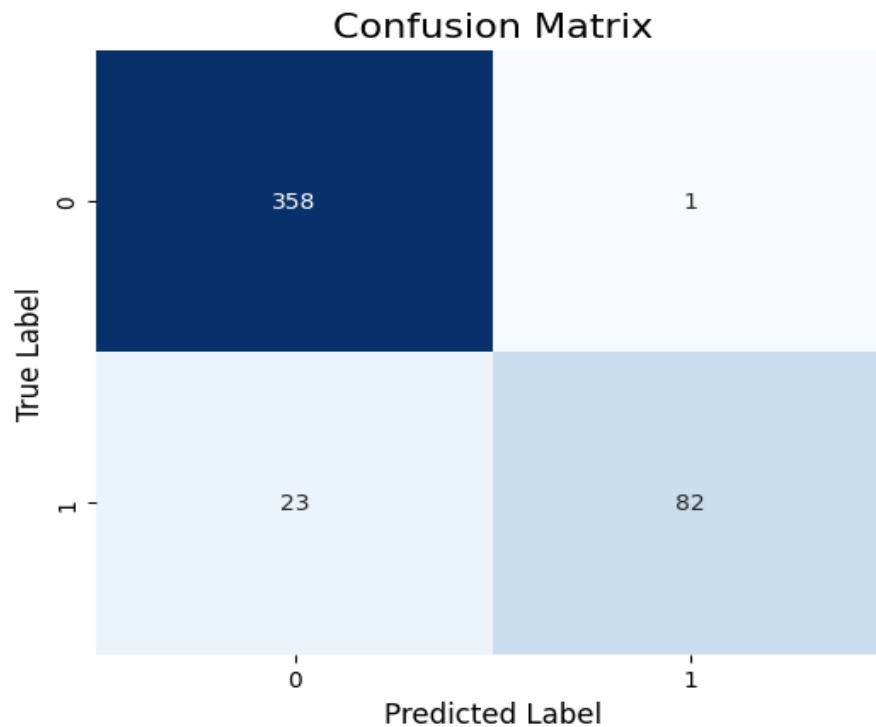
**Confusion Matrix:**

Fig 9.5: Confusion matrix values

**9.2 Comparison with Traditional Fraud Detection Methods**

In this section, we compare the performance of the Random Forest model with traditional fraud detection methods, such as rule-based systems, manual auditing, and simpler machine learning algorithms (e.g., logistic regression). The comparison highlights the strengths and weaknesses of the Random Forest approach, providing a broader context for evaluating its effectiveness.

**Rule-Based Systems**

Rule-based fraud detection systems rely on predefined thresholds and simple rules to flag potential fraud. For instance, if a taxpayer's reported income exceeds a certain threshold or if deductions are unusually high, the system might flag the tax return for further investigation. These systems are easy to implement but suffer from several limitations:

- **Limited Flexibility:** Rule-based systems struggle to adapt to complex and evolving fraud tactics. They work well for detecting simple fraud schemes but are not capable of identifying novel fraud patterns or complex interactions between multiple features.
- **High False Positive Rates:** Because rule-based systems often use rigid thresholds, they may flag legitimate filings as fraudulent, leading to high false positive rates and unnecessary investigations.

In contrast, the Random Forest model uses an ensemble of decision trees, allowing it to handle complex, non-linear relationships between features and improve its ability to detect sophisticated fraud patterns. The model's higher precision and recall values indicate that it performs better than rule-based systems in terms of both reducing false positives and correctly identifying fraud.

### **Manual Auditing**

Manual audits have been the traditional method of identifying tax fraud. Auditors examine tax returns, scrutinize discrepancies, and manually detect fraud patterns. However, this process is time-consuming, resource-intensive, and prone to human error. The key drawbacks of manual auditing include:

- **Slow Processing:** Given the sheer volume of tax returns that need to be reviewed, manual auditing is inefficient, leading to delays in fraud detection and prevention.
- **Human Error:** Auditors may miss subtle fraud patterns due to fatigue, cognitive biases, or insufficient expertise.

By automating the fraud detection process, the Random Forest model significantly outperforms manual auditing in terms of speed and accuracy. While human oversight remains important for validating predictions and investigating flagged cases, the model serves as a powerful tool for assisting auditors by quickly identifying potentially fraudulent tax filings.

## **9.3 Limitations**

Despite its impressive performance, the Random Forest model does have some limitations that need to be addressed for real-world applications. These limitations include:

### **Class Imbalance**

Income tax fraud cases are relatively rare compared to legitimate tax returns, creating an imbalance in the dataset. Although techniques like SMOTE and cost-sensitive learning were employed to mitigate this issue, the model may still struggle to detect rare fraud cases. This imbalance leads to a higher risk of false negatives (missed fraud cases) and could undermine the model's effectiveness in identifying all instances of fraud.

### **Interpretability**

One of the main challenges of using Random Forest and other ensemble methods is the lack of interpretability. The model operates as a "black box," meaning that it is difficult to understand the exact reasoning behind a prediction. In the context of tax fraud detection, where transparency and accountability are crucial, this lack of interpretability can be a significant

barrier to adoption. Tax authorities may hesitate to use the model for decision-making without clear explanations of how predictions are made.

### **Data Quality and Completeness**

The performance of the Random Forest model is highly dependent on the quality and completeness of the data. Missing or inaccurate data, such as incomplete tax filings or incorrect deductions, can negatively impact the model's predictions. Additionally, the model does not incorporate unstructured data, such as social media activity or third-party reports, which could offer valuable insights for detecting fraudulent behavior.

### **Scalability**

While Random Forest can handle large datasets better than simpler models, it still faces scalability challenges as the volume of tax data continues to increase. The computational complexity of training multiple decision trees in parallel could lead to slower processing times for large-scale datasets. Future improvements in distributed computing and parallel processing may be necessary to address this issue.

### **Adaptation to New Fraud Tactics**

Fraud tactics constantly evolve, and the model must be retrained periodically to keep up with these changes. While Random Forest can adapt to some extent, it lacks the ability to learn dynamically from new data in real-time. This limitation may make it less effective in detecting new and emerging types of fraud without frequent model updates.

### **Privacy and Security**

The use of sensitive taxpayer data for training and deploying the model raises privacy concerns. Ensuring that the model adheres to data protection regulations, such as GDPR, while still delivering high accuracy in fraud detection is a significant challenge. Moreover, the need for secure data handling practices must be addressed to maintain trust with taxpayers.

## CHAPTER-10

### CONCLUSION

This synthesizes the study's key findings and emphasizes the significance of AI and machine learning (ML) in the modern landscape of tax fraud detection. It also outlines potential future research directions and explores the broader applications of the methodology beyond income tax fraud detection.

This study set out to explore the potential of using machine learning, specifically the Random Forest algorithm, in detecting income tax fraud. The results demonstrate that the Random Forest algorithm is highly effective in identifying fraudulent tax filings from legitimate ones. Through comprehensive evaluation, the model exhibited strong performance across key metrics such as accuracy, precision, recall, F1-score, and AUC-ROC, showcasing its ability to handle complex, high-dimensional financial data. The model's ability to handle large datasets, manage intricate feature interactions, and generalize effectively was proven, making it an ideal tool for tax fraud detection. While precision was high, recall remained somewhat lower due to the inherent imbalance in fraud cases compared to legitimate filings, which is typical in fraud detection tasks. Despite this, the F1-score, which balances both precision and recall, confirmed that the Random Forest model performs well overall. Furthermore, the model showed resilience to overfitting and underfitting, indicating its robustness for real-world applications.

In comparison to traditional fraud detection methods such as rule-based systems and manual auditing, Random Forest demonstrated superior performance. Rule-based systems were found to be rigid, with high false positive rates and an inability to detect evolving fraud patterns. Manual audits, while still necessary for validation, were far slower and more resource-intensive than an automated machine learning model. Thus, the Random Forest model not only outperforms traditional techniques in terms of speed and accuracy but also offers a scalable solution for tackling tax fraud.

#### **Potential Applications Beyond Tax Fraud**

The methodology developed for income tax fraud detection can be extended to a wide variety of other domains, demonstrating the broader potential of AI-driven fraud detection systems.

1. **Banking Fraud:** Financial institutions face a constant threat from fraud, whether in the form of credit card fraud, loan defaults, or money laundering. The Random Forest model could be adapted to detect fraudulent transactions by analyzing transaction



histories, account behavior, and patterns of financial activity. By applying similar techniques to banking fraud, financial institutions can identify suspicious transactions more efficiently and reduce the risk of fraud.

2. **Insurance Fraud:** Insurance fraud, whether in the form of false claims or inflated medical expenses, is a significant issue for the insurance industry. The methodology used in this study can be applied to detect fraud in insurance claims by analyzing claims data, customer behavior, and policy details. By utilizing machine learning, insurance companies can flag suspicious claims and minimize the financial losses due to fraud.
3. **Corporate Tax Evasion:** Corporate tax evasion involves complex schemes that are harder to detect compared to individual tax fraud. The use of machine learning models like Random Forest can help tax authorities track corporate financial behavior and identify anomalies that suggest tax evasion. This could include analyzing discrepancies in reported revenues, deductions, and expenses, as well as cross-referencing data with third-party sources.
4. **Healthcare Fraud:** Healthcare fraud, including fraudulent billing, overcharging, or identity theft, is another area where AI and machine learning could play a significant role. The model could be applied to detect patterns of abuse in insurance claims, medical billing, and treatment records, ultimately reducing healthcare costs and improving service delivery.

Future enhancements for the proposed income tax fraud detection system include leveraging advanced machine learning techniques and exploring deep learning architectures like Neural Networks, Long Short-Term Memory (LSTM), or Convolutional Neural Networks (CNNs) for improved accuracy and adaptability. Incorporating ensemble methods, such as XGBoost, LightGBM, or hybrid models combining multiple classifiers, can further enhance predictive performance by mitigating biases and improving generalization on unseen data. To strengthen the system's predictive capabilities, integrating comprehensive datasets is essential. Future work can include unstructured and semi-structured data sources such as digital payment records, third-party reports, social media activity, and publicly available business data to capture multi-dimensional fraud patterns. By integrating cutting-edge technologies and addressing key challenges like privacy, scalability, and adaptability, the proposed system can become a powerful tool in combating financial fraud and upholding the integrity of the global financial system.

## REFERENCES

- [1] Enhanced Income Tax Fraud Detection System Using Machine Learning TY - BOOK;AU - Rani, Rm;AU - Anand, Amrit;AU - Agarwal, Pratham;AU - Srivastava, Ayush;PY - 2024/04/16
- [2] A. Z. Adamov, "Machine Learning and Advanced Analytics in Tax Fraud Detection," 2019 IEEE 13th International Conference on Application of Information and Communication Technologies (AICT), Baku, Azerbaijan, 2019, pp. 1-5, doi: 10.1109/AICT47866.2019.8981758. keywords: {Data Analytics;Machine Learning;Big Data;Hadoop;Taxation},
- [3] Sadgali, I., Sael, N., & Benabbou, F. (2019). Performance of machine learning techniques in the detection of financial frauds. *Procedia Computer Science*, 148, 45–54. doi:10.1016/j.procs.2019.01.007
- [4] Shweta S. Borkar, Dr. Latesh Malik, (2015). "A Survey of Fraud Detection Techniques in Financial Domain"
- [5] Kanapickienė, R., & Grundienė, Ž. (2015). The Model of Fraud Detection in Financial Statements by Means of Financial Ratios. *Procedia - Social and Behavioral Sciences*, 213, 321–327. doi:10.1016/j.sbspro.2015.11.545
- [6] B. Rajesh Kumar, V. V. R. Raju (2018). "Fraud Detection in Banking Transactions using Machine Learning Algorithms"
- [7] Maria Gonzalez; Javier Martinez; Elena Fernandez(2022), Deep Learning Approaches for Tax Fraud Detection: A Review
- [8] Amit Kumar Tyagi, Dr. Y. P. Singh (2018). "A Comparative Analysis of Data Mining Techniques in the Detection of Fraudulent Activities"
- [9] Niharika Kaul, Dr. J. L. Rana (2019). A Review on Fraud Detection using Machine Learning Algorithms"
- [10] N. Visitpanya and T. Samanchuen, "A Machine Learning Approach to Identifying Suspicious Tax Evasion Behavior in Public Financial Data," 2023 8th International Conference on Business and Industrial Research (ICBIR), Bangkok, Thailand, 2023, pp. 1152-1158, doi: 10.1109/ICBIR57571.2023.10147479.
- [11] Muhammad Rahman; Sarah Patel; Aisha Khan (2023) . "Feature Selection Techniques in Tax Fraud Detection: A Survey"
- [12] A Machine Learning-based System for Financial Fraud Detection  
TY - BOOK;AU - Andrade, João;AU - Paulucio, Leonardo;AU - Paixão, Thiago;AU

- Berriel, Rodrigo;AU - Carneiro, Teresa Cristina;AU - Carneiro, Raphael;AU - De Souza, Alberto;AU - Badue, Claudine;AU - Oliveira-Santos, Thiago;PY - 2021/11/29;DO - 10.5753/eniac.2021.18250

[13] Assessing the Role of Artificial Intelligence (AI) on Tax Fraud Detection

TY - CHAP;AU - Radhi, Wael;AU - Hamdan, Allam;AU - Binsaddig, Ruaa;PY - 2024/08/31;SP - 359;EP - 364;SN - 978-3-031-62101-7;DO - 10.1007/978-3-031-62102-4\_30

[14] AI-Driven Approaches for Real-Time Fraud Detection in US Financial Transactions: Challenges and Opportunities TY - JOUR;AU - Bello, Oluwabusayo;AU - Ogundipe, Abidemi;AU - Mohammed, Damilola;AU - Folorunso, Adebola;AU - Alonge, Olalekan;AU - Bello, Citation;PY - 2023/01/01;SP - 84;EP - 102;DO -10.37745/ejcsit.2013/vol11n684102

## **APPENDIX-A**

### **PSEUDO CODE**

#### **Step 1: Initialize SQLite Database**

1. Connect to SQLite Database.
2. Create a table tax\_records with the following fields:
  - id (Auto-incremented Primary Key)
  - first\_name, last\_name (User Names)
  - DOB
  - income, deductions, expenses, tax\_paid (Numeric inputs)
  - is\_fraud (Fraud flag, 0 or 1)
  - amount\_to\_be\_paid (Tax discrepancy amount)
3. Commit changes and close the database connection.

#### **Step 2: Define Tax Calculation Function**

1. Input: taxable\_income (income - deductions).
2. Apply tax slabs:
  - If  $\text{income} \leq 1,00,000 \rightarrow \text{Tax} = 0$ .
  - If  $\text{income} > 100,000$  and  $\leq 300,000$ : Tax = 5% on income above 100,000
  - If  $\text{income} > 300,000$  and  $\leq 700,000$ : Tax = 5% on income between 100,000 and 300,000, and 10% on income above 300,000
  - If  $\text{income} > 700,000$ : Tax = 5% on income between 100,000 and 300,000, 10% on income between 300,000 and 700,000, and 15% on income above 700,000Return calculated tax.

#### **Step 3: Train Random Forest Classifier**

1. Load dataset from tax.csv.
2. Add computed columns:
  - $\text{taxable\_income} = \text{income} - \text{deductions}$ .
  - $\text{expected\_tax} = \text{calculate\_tax}(\text{taxable\_income})$ .
3. Label fraud:
  - $\text{is\_fraud} = 1$  if  $\text{tax\_paid} < 80\%$  of  $\text{expected\_tax}$ .
4. Prepare training data:

- Features: income, deductions, expenses, tax\_paid, expected\_tax.
  - Target: is\_fraud.
5. Split data into training and testing sets (70%-30% split).
  6. Train a Random Forest Classifier:
    - Save the trained model to fraud\_detection\_model.pkl.
  7. Evaluate model using:
    - Accuracy, Precision, Recall, and F1-Score.
  8. Print the metrics.

#### **Step 4: Flask Routes**

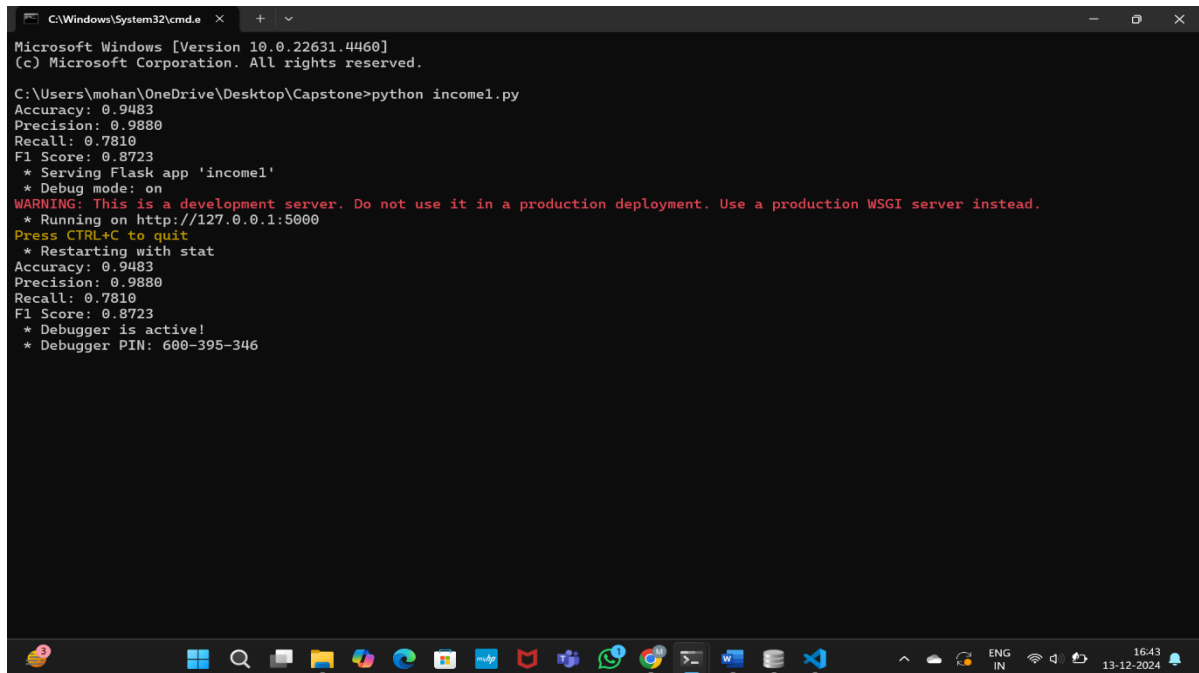
1. Home Route (/):
  - Render the home page (index.html).
2. Fraud Prediction API (/predict, POST method):
  - Input: JSON data (first\_name, income, deductions, expenses, tax\_paid).
  - Compute:
    - taxable\_income = income - deductions.
    - expected\_tax = calculate\_tax(taxable\_income).
  - Use the saved Random Forest model to predict fraud (is\_fraud).
  - Calculate amount\_to\_be\_paid = expected\_tax - tax\_paid if fraud detected.
  - Save user data and prediction results into the database.
  - Return the prediction (JSON response).
3. Generate Confusion Matrix (/confusion-matrix, GET method):
  - Load actual (y\_test) and predicted (y\_pred) labels from confusion\_data.csv.
  - Compute confusion matrix.
  - Plot confusion matrix using Seaborn heatmap.
  - Save the matrix as confusion\_matrix.png.

#### **Step 5: Run Flask Application**

- Run the Flask app in debug mode.

## APPENDIX-B

### SCREENSHOTS



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mohan\OneDrive\Desktop\Capstone>python income1.py
Accuracy: 0.9483
Precision: 0.9880
Recall: 0.7810
F1 Score: 0.8723
* Serving Flask app 'income1'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
Accuracy: 0.9483
Precision: 0.9880
Recall: 0.7810
F1 Score: 0.8723
* Debugger is active!
* Debugger PIN: 600-395-346
```

Fig 12.1 Initialization of flask web application

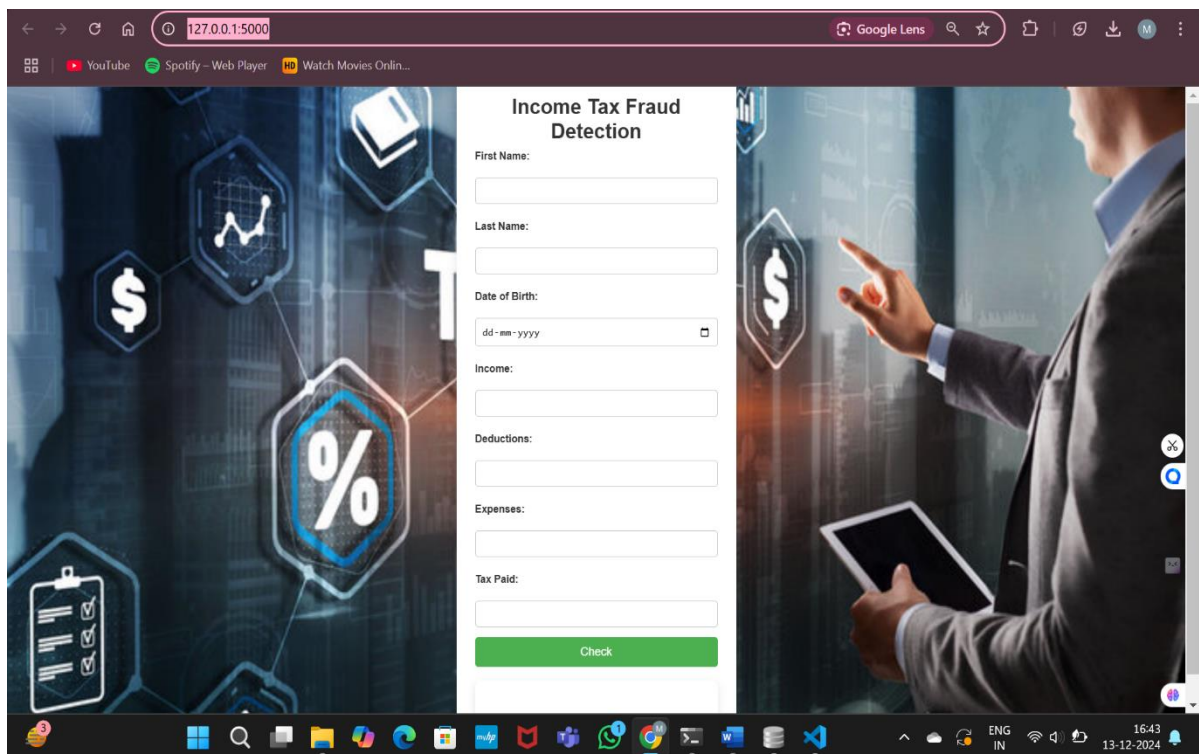


Fig 12.2 Frontend of the model

First Name:

Last Name:

Date of Birth:

Income:

Deductions:

Expenses:

Tax Paid:

Fraud Detected! Amount to be paid: 16400

Fig 12.3 Output

In Fig 12.3, the inputs are filled with the required values and the required result ie. Fraud is predicted

```

C:\Windows\System32\cmd.e  X  +  v

Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mohan\OneDrive\Desktop\Capstone>python income1.py
Accuracy: 0.9483
Precision: 0.9880
Recall: 0.7810
F1 Score: 0.8723
* Serving Flask app 'income1'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production de
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
Accuracy: 0.9483
Precision: 0.9880
Recall: 0.7810
F1 Score: 0.8723
* Debugger is active!
* Debugger PIN: 600-395-346

```

Fig 12.4 Values of the evaluation metrics

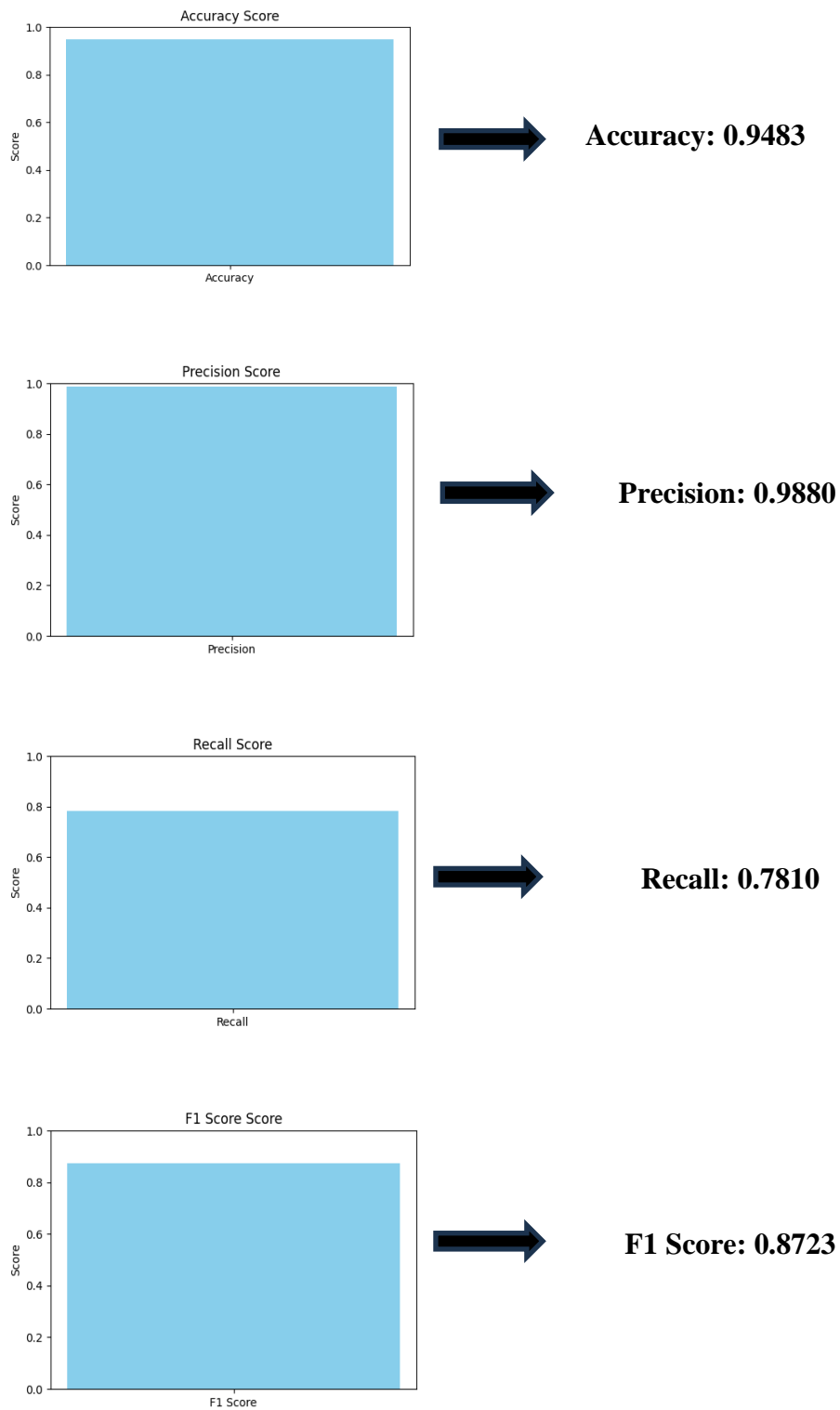


Fig 12.5 Representation of the evaluation metrics



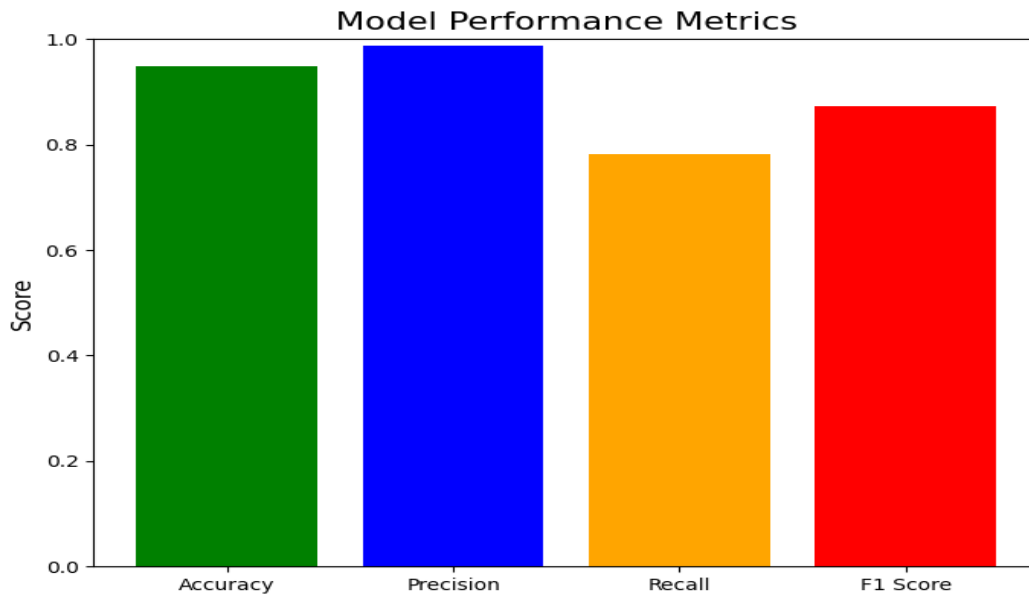


Fig 12.6 Pictorial representation of the metrics

A screenshot of the DB Browser for SQLite application. The main window displays a table named "tax\_records" with the following data:

ID	first_name	last_name	income	deductions	expenses	tax_paid	is_fraud	amount_to_be_paid
1	M	Shetty	150000.0	10000.0	60000.0	20000.0	0	0.0
2	M	Shetty	150000.0	10000.0	60000.0	15000.0	1	11000.0
3	Ram	R	145000.0	5000.0	50000.0	18000.0	0	0.0
4	Ram	R	145000.0	5000.0	50000.0	16000.0	1	10000.0
5	Ram	R	145000.0	5000.0	50000.0	18000.0	0	0.0
6	Mohan	R Shetty	180000.0	10000.0	75000.0	30000.0	0	0.0
7	Raj	M	198000.0	10000.0	100000.0	30000.0	0	0.0
8	Raj	M	198000.0	10000.0	100000.0	25000.0	0	0.0
9	Raj	M	198000.0	10000.0	100000.0	25000.0	0	0.0
10	Raj	M	198000.0	10000.0	100000.0	24000.0	1	16400.0

The interface also shows a right-hand pane for editing database cells and a bottom status bar with system information.

Fig 12.7 Database Entries

## APPENDIX-C

### ENCLOSURES

#### 1. Journal publication/Conference Paper Presented Certificates of all students.



Mohan R Shetty <mohanshetty9@gmail.com>

---

**Your submission to Mach. Learn.: Sci. Technol.: MLST-103251**

---

**Machine Learning: Science and Technology** <onbehalfof@manuscriptcentral.com>

Fri, 3 Jan, 11:22 AM

Reply to: <mlst@iopublishing.org>

To: <mohanshetty1@gmail.com>, <mohanshetty9@gmail.com>, <amulyassathish@gmail.com>, <kksuchithra414@gmail.com>, <sumanthtch@gmail.com>, <vineetha.b@presidencyuniversity.in>

Dear Mr Shetty,

Re: Income Tax Fraud Detection Using AI&ML

Manuscript reference: MLST-103251

Thank you for submitting your Paper to Machine Learning: Science and Technology.

The reference number for your manuscript is MLST-103251. Please quote this whenever you contact us about the manuscript.

To track the progress of your article, please visit our [Publishing Support website](#) and enter your manuscript ID as directed. If you use [WeChat](#), you can go to the article tracking service in the official IOP Publishing WeChat account.

#### Using your Author Centre

You can log into your Author Centre at <https://mc04.manuscriptcentral.com/mlst-iop>. Once you are signed in, you will be able to:

- Follow the progress of your manuscript
- Read the reviewer reports
- Send us your electronic files

#### Important publishing information

At Machine Learning: Science and Technology, we make manuscripts available to readers on the journal website within 24 hours of acceptance.

This means that, unless you have opted out, the accepted version of your manuscript will be visible before it is proofread and formatted to our house style.

If you are planning any press activity or you are engaging in any IP or patent application, you may prefer your work not to be published immediately.

If this is the case, and you have not already opted out during the submission process, please let us know as soon as possible.

Yours sincerely,

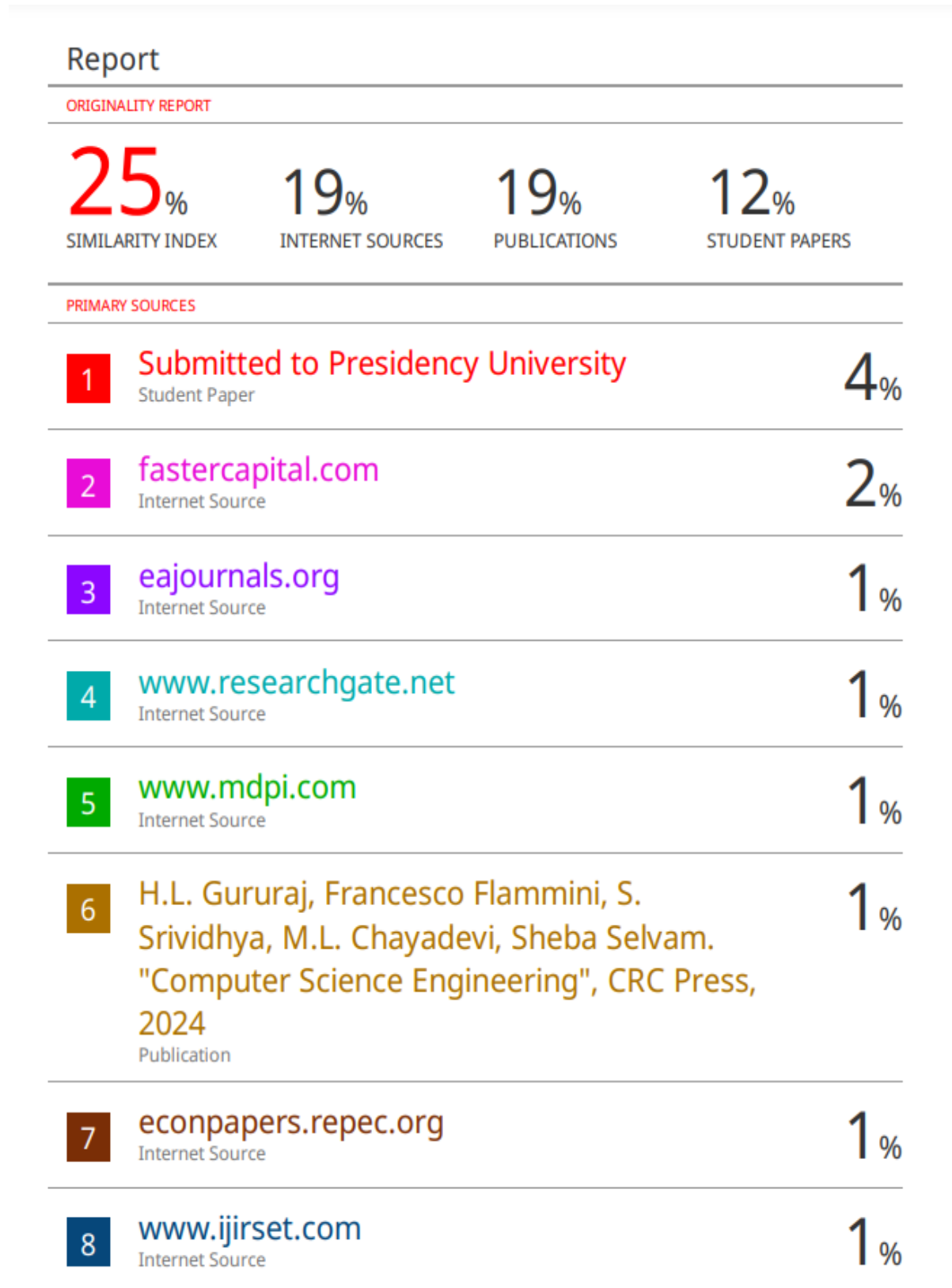
On behalf of:  
Machine Learning: Science and Technology

Editor-in-Chief: Kyle Cranmer

[iopscience.org/mlst](http://iopscience.org/mlst) | [mlst@iopublishing.org](mailto:mlst@iopublishing.org)

Impact Factor: 6.3

## 2.Similarity Index / Plagiarism Check report clearly showing the Percentage (%)



### **3.Sustainable Development Goals (SDGs) Mapping**

Sustainable Development Goals (SDGs), which are an urgent call for action by all countries - developed and developing - in a global partnership. They recognize that ending poverty and other deprivations must go hand-in-hand with strategies that improve health and education, reduce inequality, and spur economic growth – all while tackling climate change and working to preserve our oceans and forests. SDG 17 goals.

Our model name “Income Tax Fraud Detection Using AIML” is mapped and related to the SDG 16 goal (78%).

SDG 16 refers to “Peace, Justice, and Strong Institutions”. SDG 16 focuses on promoting peaceful, inclusive societies, ensuring access to justice, and building accountable and transparent institutions. The development of this model results in promoting justice, strengthening Institutions and government can ensure that public pays the necessary amount that should be paid. Income tax fraud undermines the integrity of financial systems, reduces government revenues, and hampers the ability to fund public services like healthcare, education, and infrastructure.

This model also indirectly contributes to SDG 8 which refers to “Decent Work and Economic Growth”.