Advancing Income Tax Fraud Detection: A Comprehensive Review of Machine Learning and Deep Learning Models

Shashank S N[1], Abhin K M[2], Srihari A S[3], Shadakshari D[4], Dr. Senthil Kumar Swami Durai[5]
[1,2,3.4]B. Tech Computer Science Engineering, Presidency University, Bangalore
[5]Professor & HOD CSE, Presidency University, Bangalore

***Abstract*—As digital financial systems continue to expand and transaction volumes rise, the risk of fraud has increased considerably. This paper explores how Machine Learning (ML) and Deep Learning (DL) techniques can be utilized to detect income tax fraud, emphasizing important metrics like accuracy, precision, recall, and F1-score to assess the performance of the models.**

Core Methodologies:

The model applied is those of the random forest, XGBoost and Decision Tree Classifier, together with CNN and LSTM architectures in deep learning models. In addition to these improvements in models, preprocessing is done that involves feature engineering, using one-hot encoding of categorical features, and then normalization. The study focuses on a highly imbalanced synthetic dataset with a fraud rate of 0.9% and 1 million entries, tackling issues related to computational complexity and class imbalance with customized strategies.

Performance Insights:

Random Forest and XGBoost proved to be better since the F1-scores obtained were 0.9472 and 0.9522, respectively. However, for the same experiment, the performance of Decision Tree classifiers was competitive with an F1-score of 0.9407 and relatively very less computation time, and for CNNs and LSTMs, it could not be impressive as there is less recall and relatively higher computation, and hence F1-scores 0.3341 and 0.5110 respectively.

This paper discusses the strengths and weaknesses of each model in fraud detection, discussing the implications of imbalanced datasets and computational trade-offs. It provides actionable insights for scholars, policymakers, and industry stakeholders who are looking to improve financial stability through advanced AI-driven fraud detection systems. Future recommendations include optimizing DL architectures for recall and leveraging ensemble approaches to further improve detection accuracy and efficiency.

***Index Terms*—Machine Learning, Deep Learning, Fraud Detection, Income Tax, Financial Security, Random Forest, XGBoost, LSTM, CNN, Imbalanced Data**

I. INTRODUCTION

A Background:

This is in addition to the rapid increase in the volume of transactions, which makes both security assurance and fraud prevention here quite challenging. Despite some degree of effectiveness to these fraud detection systems in most cases, they are sometimes handicapped in dealing with modernity, which has a whole complex web of financial transactions whose schemes are becoming so refined and sophisticated and thus not so easy to spot out. In this scenario, ML and DL are disruptive technologies that can potentially revolutionize the approach followed in the process of fraud detection. The fraud detection systems apply the techniques of ML and DL for scanning enormous datasets of transactions in order to locate anomalies and patterns that can otherwise not be found.

These automatically improve the decision-making process by minimizing human efforts to provide fast and accurate fraud detection. For example, ensemble methods, like Random Forest and boosting, have been proved highly effective in detecting fraud in financial statements, with certain cases achieving accuracy above 90%. These methods combine multiple models to improve robustness and accuracy [1]. Deep learning models are well-suited for detecting complex and new fraud scenarios because they automatically extract features from raw data. The three stages of the ML process are data preprocessing, model building, and evaluation. Application of these stages in fraud detection helps overcome the dynamic nature of fraud patterns for financial institutions. Likewise, DL models also do very well in extracting useful information from raw transaction data for strong anomaly detection. This feature is especially true in cases where fraudsters invent new techniques to avoid traditional security apparatuses.

For the sake of experimentation, researchers have explored datasets with different sizes, such as one study that used 10 lakh transactional entries to benchmark the performance of ML and DL models in fraud detection tasks. Such synthetic datasets become a critical foundation for testing the scalability and effectiveness of fraud detection algorithms [1]. Advanced AI and ML technologies can improve the defenses of the financial sector, which would allow it to be more efficient and reliable in detecting fraudulent activities.

B. Objectives:

The current study evaluates the effectiveness of various ML and DL models in the context of fraud detection, especially their applications in income tax fraud detection. Through an extensive review, this study will attempt to fulfill the following:

- Performance evaluation of different machine learning models such as Random Forest, Decision Trees, XGBoost, and DL models like Convolution Neural Networks and LSTMs for fraudulent transactions.
- Comparing the performance of these models and evaluating them using necessary measures such as precision, recall, accuracy, and F1-score to ascertain which model performs best.
- Analyze the transaction patterns and anomalies for the models, including information about basic behaviors associated with fraud.
- Steps required: Analyze the advantages and disadvantages of ML and DL techniques in fraud detection to inform their adoption in an actual financial system. Propose possible avenues towards enhancing ML and DL-based fraud detection methods and areas that need further research and innovation in this field.

II. REVIEW EXISTING WORK

The area of income tax fraud detection is one where machine learning, deep learning, and artificial intelligence techniques are more and more being applied to improve the accuracy and efficiency of detection. This review combines the methodologies adopted in various studies, especially focusing on those discussed in the base paper [1] and integrating findings from other major works in the field.

A. Supervised Learning Approaches

Supervised learning is used most frequently for income tax fraud detection. Various researchers, in different studies, established the feasibility of Logistic Regression, Decision Trees, and even Support Vector Machines (SVM) to classify between false and valid tax returns. While referring to the base paper, it can be noted that features play a significant role in supervised models. Irrelevant or redundant features lower the accuracy of the models to a large extent [1]. Other researchers also express that supervised learning algorithms are already used to great advantage in related areas such as fraudulent detection in blockchains [2]. An additional application includes ANNs being applied for fraudulent activities, attaining a classification accuracy of 92% by employing deep learning methods to enhance models' performance [3].

B. Unsupervised Learning Techniques

In cases where labeled data is not available, unsupervised learning methods, like anomaly detection, are good tools for identifying suspicious activity. The base paper shows the promise of unsupervised learning, especially clustering and anomaly detection techniques, in the analysis of financial data [1]. A study on under-reporting fraud demonstrates the use of autoencoders for anomaly detection and how unsupervised methods can label previously undetected fraudulent activities without the need for historical labeled data [6]. Other studies also investigate unsupervised methods for tax fraud detection, emphasizing that a combination of both supervised and unsupervised techniques is needed to make fraud detection more robust [5].

C. Hybrid and Ensemble Methods

Hybrid methods that combine supervised and unsupervised learning have gained significant attention in recent years due to their ability in enhancing the accuracy and robustness of fraud detection systems. The base paper argues for hybrid models that incorporate both kinds of learning so that the system may make use of labeled data and discover the unknown patterns in unlabeled data [1]. One of the studies integrates boosting algorithms, like AdaBoost and Gradient Boosting, with traditional classifiers, such as decision trees, to enhance fraud detection in income tax filings [4]. Another framework integrates XGBoost and autoencoders into a multi-module system, improving the performance of detection through classification and anomaly detection [5].

D. Deep Learning Approaches

Deep learning, especially through ANNs, is gaining pace in fraud detection because it can deal with large complex datasets. Base Paper This base paper stresses that deep learning models, particularly ANNs, are more effective in detecting fraud patterns which the traditional machine learning model would have missed, but such models require large datasets and massive computational resources [1]. The application of a Multilayer Perceptron (MLP) neural network in one study to detect tax fraud based on personal income tax returns also shows the benefits of deep learning, which achieved an accuracy of 84.2% [8]. Hyperparameter tuning, such as batch size and the number of hidden layers, is also necessary for improving model performance [3].

E. AI and Big Data Analytics

The systems of tax fraud detection are being transformed by AI and big data analytics, which provides the capacity to process and analyze massive datasets in real time. The base paper describes how AI technologies, particularly big data analytics, can be useful for tax authorities in dealing with large amounts of data to reveal fraud patterns that might otherwise remain hidden [1]. Other research studies on using machine learning and AI technologies in finance focus on these technologies because they enhance capabilities in detecting frauds through real-time monitoring of transactional data, ensuring a faster and effective detection system [10]. Similarly, AI-based detection systems employing predictive analytics are highlighted to have the ability of preventing fraud before it can happen to result in better efficiencies in the operations and mitigate losses incurred [7].

F. Challenges and Future Directions

Although huge strides have been made into AI-driven fraud detection, there are still many more challenges to be addressed. This base paper points out concerns of model interpretability, more so in deep learning models acting like a "black box," which makes the audit and regulatory processes hard to fathom or trust the model's results [1]. This has been further emphasized in some studies that call for a need for more interpretable models to enhance transparency and regulatory acceptance [9]. The constantly changing nature of tax fraud schemes requires that models be continuously retrained, a challenge that must be overcome to ensure that fraud detection systems are effective in the long term.

III. METHODOLOGY

A. Dataset Description:

This study uses a dataset of 109,066 records of transactions with 14 features. These features help bring out all kinds of aspects within the transactions. Data Both numerical and categorical columns are presented for information such as a particular transaction, account balance, and so on, while providing contextual details.

The numerical columns consist of the transaction amount, the old and new balance of both origin and destination accounts, and a noise feature added to test the model's robustness. The categorical columns are comprised of transaction types such as PAYMENT, CASH_OUT, DEBIT, and CASH_IN; it further has the transaction time categorized into morning, afternoon, evening, etc.; and the transaction location like EU, Africa. The main target variable is the binary label is Fraud indicating if the transaction was fraudulent (1) or not (0).

The dataset is a bit imbalanced, with only a few percent of fraudulent transactions compared to the number of legitimate transactions. There are a few missing entries in columns like nameDest, oldbalanceDest, newbalanceDest, and isFraud, which were filled during preprocessing. However, the dataset is robust enough to explore the application of machine learning and deep learning models for fraud detection.

The size of this dataset is about 11.6 MB, making it decent for experimenting with lots of models and offering complexity adequate to test how various techniques are able to handle large-scale fraud detection tasks. Each record will hold detailed transactional information suitable for identifying patterns for helping detect fraudulent behavior in real-world financial transactions. The following table shows a few rows of the dataset as presented, showing the variety in terms of transaction types, amounts, and locations.

B. Models Used:

In this paper, five different models—Random Forest, Decision Tree, XGBoost, Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM)—are used to detect fraudulent transactions. These models were chosen because of their proven effectiveness in both traditional machine learning and deep learning tasks and their ability to handle the complexities of fraud detection in financial datasets.

- Random Forest:

Random Forests is an ensemble learning method that is commonly used for performance prediction because it can handle large and complex datasets. By aggregating predictions from multiple decision trees, Random Forests enhance robustness and reduce overfitting in predictions, which is important when dealing with varied learner data. This, in the context of predicting student success, educational data studies have proven that Random Forests can capture relevant features such as prior academic performance, engagement levels, and socio-economic factors [11]. Another feature that helps improve the predictive power of the model is feature selection; it ensures that only the most relevant student data are used in making decisions. Because Random Forests is very efficient and can deal with the nonlinear relationship of data, it has become a great predictor of academic outcomes [12].

- Decision Tree:

Decision Trees are a popular method of learner performance prediction again due to interpretability. In educational contexts, one can classify students based on different risk categories related to their behavior in academics by using decision trees. Decision trees are interpretive and understandable, suitable for educational settings in which interpreting the decision-making process matters [17]. They tend to overfit, especially in noisy or complex data. Techniques such as pruning and boosting are usually applied to improve the generalization ability of the model. Decision Trees are particularly useful in settings where understanding the decision-making process is important for educational administrators, as they provide clear and interpretable rules for predictions. Such a plain structure helps to pinpoint the variables that affect academic performance, providing useful information regarding study habits, class participation, and time management, among others [13]. In educational usage, this model ensures transparency in predicting learner outcomes.

- XGBoost:

XGBoost has achieved much attention due to scalability and handling large data while delivering high accuracy. It is widely applied in machine learning competitions and real-world applications, including predictive learner performance. The XGBoost uses a gradient boosting framework incorporating regularization to optimize the accuracy of the model while avoiding overfitting. In summary, studies have shown the performance capabilities of XGBoost and have provided excellent outcomes as applied in educational environments where datasets could be both extensive and broad. This model can accommodate various learner data, whether their history or behavioral data, giving powerful means to predict student results. XGBoost's ability to leverage multiple data sources has made it highly accurate and effective in real-world educational settings for evaluating learner performance [14]. It is also very scalable for millions of data points, making it the most popular tool for educational data analysis projects [20].

- Convolutional Neural Network (CNN):

Even though the application of CNNs to image recognition is more prevalent, this can also be applied in analyzing educational data. CNNs are suited for recognizing patterns within the data, and hence suitable to forecast student learner's performance, especially when using such sequential data as time series concerning their behaviors and interactions. One of the advantages of applying CNN is that it can automatically obtain the features from the raw input data, thus decreasing manual feature engineering. Studies have demonstrated that CNNs can predict student success and student behavior patterns with a high level of accuracy, making them a very promising tool for predicting future performance based on past interactions and activities [15]. The use of CNNs in educational datasets allows the detection of hidden patterns that traditional methods might not capture, improving prediction outcomes [17]. The main strength of CNNs lies in applications that involve large, unstructured data such as student interaction logs in discovering the deeper relationships between actions and performance outcomes [18].

- Long Short-Term Memory (LSTM):

Long Short-Term Memory (LSTM) networks are a form of RNN, which has demonstrated its efficiency in the context of sequential data prediction and, therefore, is perfectly fit for learner performance prediction over time. LSTMs capture long-term dependencies and learn from the sequence of past student behaviors and outcomes. Therefore, tasks like predicting student dropout or long-term academic achievement are perfectly suited to LSTMs. LSTMs have been applied in educational environments to predict the development of student performance over time, thus enabling accurate predictions that inform timely interventions [16]. Since LSTMs can model long-term dependencies, they are capable of making better predictions for learner performance than other models, particularly when past data plays a large role in determining future outcomes. LSTMs also present useful representations for predicting performance in personalized learning environments, whereby the advancement of students depends on intricately unfolding patterns in time [19].

These models were selected to represent a broad spectrum of machine learning and deep learning techniques, offering both interpretable, traditional methods (Random Forest, Decision Tree) and advanced, automated methods (XGBoost, CNN, LSTM). This combination allows for a comprehensive evaluation of the strengths and weaknesses of different approaches in detecting fraudulent financial transactions. By combining both ensemble models and deep learning methods, this research aims to discover the best techniques for enhancing the precision and performance of fraud detection systems in dynamic financial environments.

C. Chunk Processing

Handling huge datasets is one of the most important challenges for most machine learning tasks, particularly for financial data, where it is quite large in number of records. Therefore, the chunk-based approach was employed in this study, dividing a big dataset into smaller pieces known as chunks that could then be fed into the model one by one. This ensures that the data will be processed iteratively with low usage of memory and computing time.

a. Chunk-Based Approach Overview:

1. Data Partitioning:

The total dataset, comprising 109,066 records, is partitioned into smaller pieces. For this experiment, the chunk size of 100,000 entries was used, which yielded roughly two chunks of data. With this partitioning of data, the system can process a small portion of the data at any given time, thus it aids in the effective management of system memory, especially when training deep learning models that consume much computational power.

2. Training on Chunks:

Each chunk is processed independently, where the training data is fed into the machine learning and deep learning models. For each chunk, the models undergo the typical pipeline stages of data preprocessing, model training, and evaluation. Preprocessing data includes encoding categorical variables, normalizing numerical features, and handling missing values. After preprocessing, the data set is split into training and testing sets. The models train on the training set and performance metrics (accuracy, precision, recall, F1-score) are evaluated on the test set for each chunk.

3. Model Aggregation:

During and after processing each chunk, results (metrics) are accumulated. For instance, classification model predictions from each chunk accumulate, and final performance metrics are calculated based on results from all chunks combined. This approach ensures models test on the full data without having to load everything at once into memory.

4. Efficiency and Scalability:

Chunk-based approach ensures the process is scalable, since it is possible to train on big datasets that would exceed the memory capacity of a single machine. The approach also improves processing time since it distributes the computation load over smaller subsets of data. This approach is more beneficial in cases where the dataset has millions of records, such as fraud detection, to avoid bottlenecks that may arise due to memory.

5. Parallelization Potential:

The chunk-based approach, besides efficiently handling large datasets, has the potential for parallelization. Training times could be reduced by training multiple chunks simultaneously, thereby improving the overall performance of the model and its assessment faster. This parallelization method may be further explored for experiments.

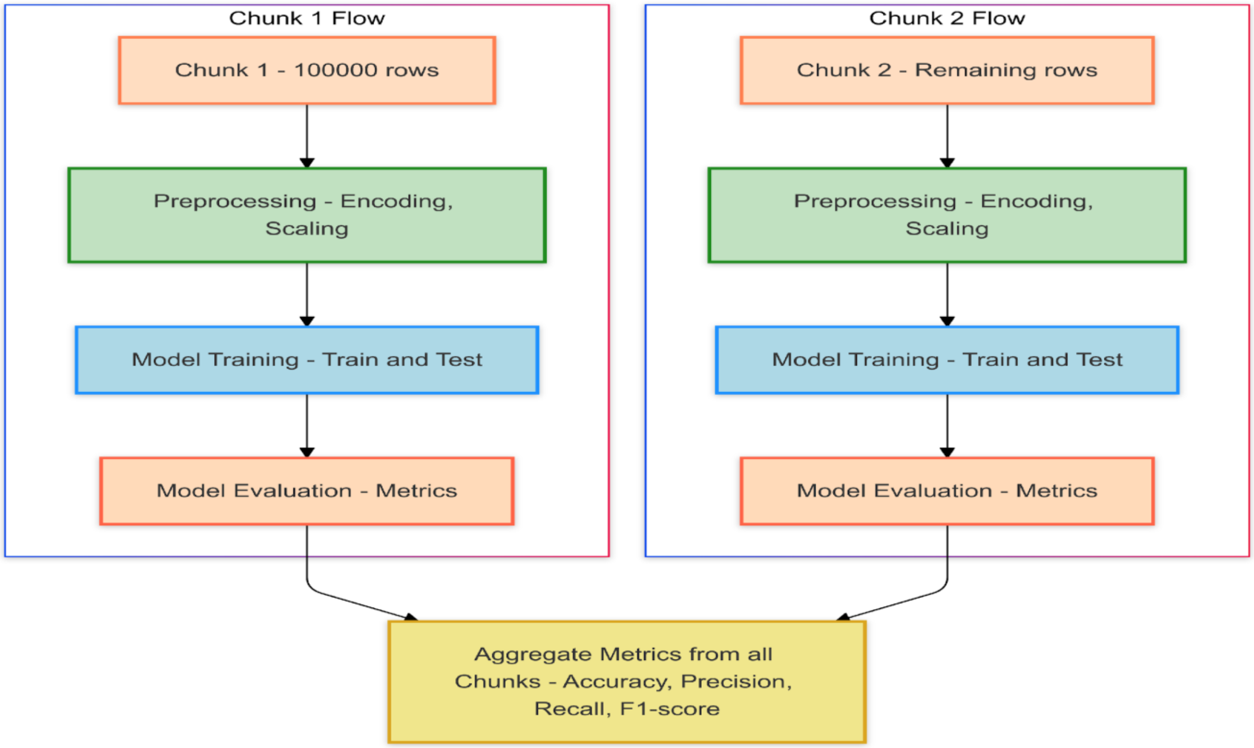
Chunk 1 Flow
Chunk 1 - 100000 rows
Preprocessing - Encoding, Scaling
Model Training - Train and Test
Model Evaluation - Metrics
Chunk 2 Flow
Chunk 2 - Remaining rows
Preprocessing - Encoding, Scaling
Model Training - Train and Test
Model Evaluation - Metrics
Aggregate Metrics from all Chunks - Accuracy, Precision, Recall, F1-score

Figure 1. Simplified diagram illustrating the chunk-based processing workflow

Such a methodology enables easy scalable model training and efficient model testing so that datasets as huge as they are processed in a manner which will not take such huge computations. Moreover, the nature of chunk processing assists memory to have appropriate management, applying deep models like LSTMs, CNNs as they wouldn't require humongous resources on computation.

D Evaluation Metrics:

To evaluate the fraud-detecting models employed in this research, we made use of several key performance metrics. The metrics are critical because they offer an understanding of how these models are balanced in terms of accuracy, precision, recall, computational efficiency, and general performance. Some of the metrics are described as follows:

- Accuracy:

Accuracy measures the percentage of correctly classified instances, fraudulent and non-fraudulent over the total number of samples. Though it gives a broad assessment of the model, its utility is highly limited to highly imbalanced datasets where the majority class dominates.

$$ACCURACY = \frac{TP + TN}{TOTAL\ INSTANCES}$$

- Precision:

Precision: Precision is the number of fraud cases detected to be true divided by the total number of cases called fraudulent. It is a very significant metric.

$$PRECISION = \frac{TP}{TP + FP}$$

- Recall:

Recall, also known as sensitivity, computes the number of correctly detected fraudulent cases over the number of fraud cases in the data set. A high recall is necessary for having minimal undetected cases of fraud.

$$RECALL = \frac{TP}{TP + FN}$$

- F1-Score:

The F1-score is a harmonic means of precision and recall, which gives a balanced view of the performance of the model. It is quite useful in imbalanced datasets where the trade-off between precision and recall needs to be carefully considered.

$$F1 - SCORE = 2 \times \frac{PRECISION \cdot RECALL}{PRECISION + RECALL}$$

- Time Taken:

Time taken measures the computational efficiency of each model, that is the time taken to perform inference on the test dataset. The metric is very important as it determines the feasibility of deployment in real-time fraud detection systems.

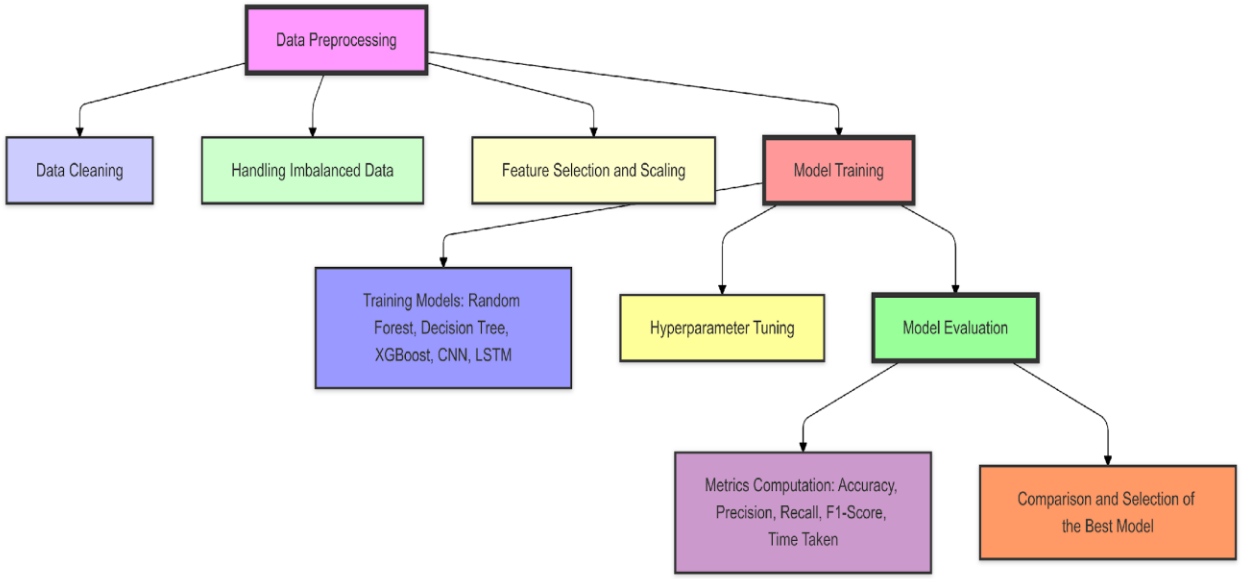
Data Preprocessing
Data Cleaning
Handling Imbalanced Data
Feature Selection and Scaling
Model Training
Training Models: Random Forest, Decision Tree, XGBoost, CNN, LSTM
Hyperparameter Tuning
Model Evaluation
Metrics Computation: Accuracy, Precision, Recall, F1-Score, Time Taken
Comparison and Selection of the Best Model

Figure 2. Flowchart illustrating the end-to-end process followed in the research:

IV. RESULTS AND DISCUSSION

This section details a comparison of various machine learning models applied for fraud detection in terms of performance. These include Random Forest, Decision Tree, XGBoost, Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM) networks.

A. Performance Metrics Comparison:

Model	Accuracy	Precision	Recall	F1-Score	Cases Detected	Time Taken (s)
Random Forest	0.9600	0.8998	0.9999	0.9472	11982	26.94
Decision Tree	0.9553	0.8996	0.9857	0.9407	11815	0.80
XGBoost	0.9639	0.9087	1.0000	0.9522	39554	3.36
CNN Model	0.7061	0.9003	0.2051	0.3341	2457	77.88
LSTM Model	0.7537	0.8921	0.3581	0.5110	4328	49.16

Table 1. Performance Metrics for All Models

B. Visual Comparisons:

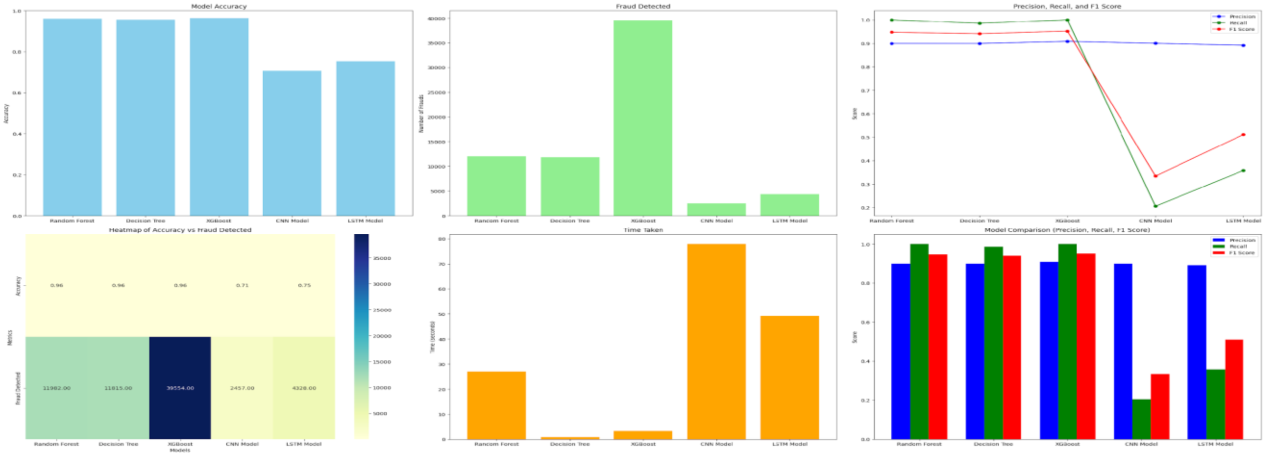

Figure 3. Model performances' evaluation graphs

The graphs evaluate the performance of income tax fraud detection models in several dimensions. The bar charts and heatmap compare accuracy, frauds detected, and time taken, thus giving clear insights into the effectiveness, efficiency, and suitability of the model for fraud detection. The precision, recall, and F1-score plots provide a deeper analysis of the reliability of the prediction by balancing the detection of frauds with minimizing false positives or negatives.

C. Discussion:

The experimental results display that XGBoost outperformed other models through all these significant performance metrics, thus clearly making it the best approach to income tax fraud detection in this experiment. XGBoost had an accuracy of 96.39%, which is the highest of all models, a precision of 0.9087, and a mean F1-score of 0.9522, which indicates its strong ability to balance precision and recall for its predictions. Furthermore, XGBoost identified the largest number of fraudulent cases (39,554) while maintaining a computation time of only 3.36 seconds, which is highly efficient compared to other models. These results highlight the advantages of gradient-boosting algorithms in structured datasets, particularly their ability to capture complex non-linear relationships and feature interactions. Additionally, XGBoost effectively handles class imbalances, as observed in this dataset, where the fraud rate is only 0.9%. Its ability to fine-tune hyperparameters such as learning rate, tree depth, and subsampling ratio makes it robust to ensure better generalization and higher accuracy.

Random Forest and Decision Tree also performed well but were a little behind XGBoost. The accuracy of Random Forest was 96.00%, the precision was 0.8998, and F1-score was 0.9472, closely challenging XGBoost. Its computation time is 26.94 seconds, although it's significantly higher because of its ensemble nature, which involves training multiple decision trees and aggregating the results. It makes it computationally expensive, particularly for large data sets with millions of entries. Decision Tree, however, was computed faster: it took 0.79 seconds to complete its task, but its accuracy was slightly lower; it was 95.53%, with an F1 score of 0.9407. Though Decision Tree is a simpler and more interpretable model, the susceptibility of overfitting restricts its generalization capability in comparison with ensemble methods such as Random Forest and XGBoost.

The performance of deep learning models, CNN and LSTM, was noteworthy with poorer accuracy and computational efficiency. CNN achieved only 70.61% accuracy and that had a recall of 0.2051 and an F1-score of 0.3341. For example, LSTM achieved 75.37% accuracy with a recall of 0.3581 and an F1-score of 0.5110, respectively, and these metrics represent how the LSTM cannot detect those necessary patterns in structured tabular data. Deep learning models have strong expertise in areas where data are not structured, like images, audio, or text data in which spatial or temporal relations are present. However, the structured datasets, such as the one in this research, demand a lot of feature engineering and domain-specific adaptation to work with deep learning. In addition, the computational time was much more significant for these models. CNN took 77.88 seconds and LSTM took 49.16 seconds, mainly because of the complexity of their architectures and training processes. Thus, they are less practical for real-time or large-scale fraud detection tasks.

A key observation is the trade-off between accuracy and computational efficiency across models. XGBoost strikes the best balance, offering high accuracy and efficiency, making it suitable for real-world deployment where both performance and scalability are critical. Random Forest provides comparable accuracy but at a higher computational cost, making it more appropriate for scenarios where processing time is not a constraint. Decision Tree, although computationally quite light, compromises in some accuracy and robustness, making it more suitable for quick, interpretable results in a time-sensitive context. CNN and LSTM, although powerful for complex data structures, require deep optimization and may not be ideal for structured datasets without great feature engineering and computational resources.

Findings indicate model selection as being task dependent. For income tax fraud detection, where accuracy and scalability are critical, XGBoost is the most viable alternative. Random Forest may be considered in reserve, when computational resources are rich; while Decision Tree can be used for less critical applications, requiring faster outputs. The performance of CNN and LSTM is so poor that a specialized adaptation or a hybrid approach is necessary to bring deep learning models up to speed with structured data analysis. Future work may include working on overcoming these limitations in the future with advanced feature extraction methods, synthetic data generation to boost training, and hybrid models that have the interpretability of decision trees but the representational power of deep learning. Moreover, bringing forward model scalability and real-time implementation should become a subject of work, as this will be crucial for deployment in practical settings.

The study also draws attention to the role of dataset characteristics in determining model performance. The imbalanced nature of the dataset, with a fraud rate of 0.9%, caused major problems for both recall and precision. XGBoost was well-handled in this respect, but deep learning approaches suffered from the lack of intense preprocessing. Future research would look into incorporating such techniques as oversampling, under sampling, or even the generation of synthetic data to combat class imbalance (such as SMOTE). In the last, it could make its way towards an extension which encompasses ensemble deep learning techniques or including some domain knowledge in the engineering of features.

V. CONCLUSION AND FUTURE WORK

A. Conclusion:

This study evaluated a number of machine learning and deep learning models for the detection of income tax fraud, based on metrics such as accuracy, precision, recall, F1-score, detected fraud cases, and computational efficiency. This study provides significant findings:

1. Best Performing Model:

XGBoost had the best accuracy of 96.39%, the highest F1 score of 0.9522, and the maximum number of fraudulent cases detected, amounting to 39,554 cases.

a. Reason for Success:

XGBoost's gradient-boosting mechanism was able to capture complex patterns in the data, thereby showing that it is robust for handling structured tabular datasets. Its relatively low computational cost of 3.36 seconds further establishes its suitability for large-scale applications.

2. Tree-Based Models as Strong Contenders:

a. Random Forest provided near-equivalent performance with accuracy of 96.00% and F1 score 0.9472, detected 11,982 fraud cases.
b. Decision Tree, though less accurate (95.53%) and with a lower F1 score of 0.9407, had the computational speed at 0.795 seconds, which proved to be beneficial for time-critical operations.

3. Limitations of Deep Learning Models:

a. The performance of CNNs and LSTMs were suboptimal at accuracy scores of 70.61% and 75.37%, respectively. The low recall values (CNN: 0.2051, LSTM: 0.3581) reveal the inability of the models to identify cases of fraud reliably.

b. Reasons for Underperformance: These models face difficulties in the structured tabular data format that it cannot handle the spatial or sequential complexity of data on which these are typically excellent. But their higher computational cost (CNN: 77.88 sec, LSTM: 49.16 sec), makes it impossible for deployment into the real-world with large datasets.

B. Limitations:

The following constraints were identified in the study:

- Data Imbalance:

Although stratified sampling was used, the nature of the dataset is very imbalanced, with a fraud rate of 0.9%, which could have affected model generalizability.

- Feature Representation:

The use of one-hot encoding for categorical variables may have resulted in loss of contextual information.

- Suitability of Models:

Deep learning architectures were not optimized for structured tabular data, and therefore, models may not be performing optimally and have higher training times.

C. Future Work:

Building on the results of this study, several directions for future research and development are proposed here:

- Hybrid Model Development:

Explore hybrid approaches combining XGBoost with deep learning architectures to leverage strengths from both methodologies.

- Addressing Imbalanced Data:

Use techniques such as oversampling like SMOTE (Synthetic Minority Oversampling Technique) or cost-sensitive learning to further enhance the fraud detection rate.

- Imbalance Mitigation:

SMOTE or cost-sensitive learning would be applied to handle imbalance.

- Model Explainability:

Add SHAP (SHapley Additive explanations) interpretability framework for model decision making, enabling transparency and more in-depth understanding of features most crucial for prediction.

- Real-Time Deployment:

Use parallelization for XGBoost, ensuring deployment on scalable and low-latency data pipelines.

D. Summary:

The present work concludes that XGBoost is the most effective model for income tax fraud detection with the highest accuracy achieved as 96.39%, precision 0.9087, and F1-score 0.9522 and efficient in detecting the maximum number of cases within the computation time of 3.36 seconds. The gradient-boosting framework it employs exhibited great robustness towards handling complex patterns in structured data and hence very apt for large-scale, real-world deployment. While the deep learning models such as CNN and LSTM performed poorly, the structured nature of the dataset did not allow them to work properly. There is also no optimization done for the domain-specific architecture. Future work would involve improving the dataset by balancing it out, feature representation with more advanced methods like embeddings, and adoption of explainability techniques such as SHAP to find out the key drivers of fraud. All this would pave the way for higher detection rates, scalability, and trustworthiness, in preparation for more holistic AI-based fraud detection systems.

REFERENCES

[1] Matin N. Ashtiani, Bijan Raahemi, *Intelligent Fraud Detection in Financial Statements Using Machine Learning and Data Mining: A SystematicLiteratureReview*.DOI:10.1109/ACCESS.2021.3096799, 2021

[2] Rohan Kumar C L, Ali Mohammed Zain, Sanjay Kumar H P, Prajwal A V, Dr. Sudarshan R, *Comparative Study of Machine Learning Algorithms for Fraud Detection in Blockchain*. DOI: 10.48175/IJARSCT-5474, 2022

[3] Belle Fille Murorunkwere, Origene Tuyishimire, Dominique Haughton and Joseph Nzabanita, *Fraud Detection Using Neural Networks: A Case Study of Income Tax*. DOI: 10.3390/fi14060168, 2022

[4] Dr RM Rani, Amrit Anand, Pratham Agarwal, Ayush Srivastava, *Enhanced Income Tax Fraud Detection System Using MachineLearning*, DOI:10.13140/RG.2.2.25755.68648,2024

[5] N. Alsadhan, *A Multi-Module Machine Learning Approach to Detect Tax Fraud*. DOI: 10.32604/csse.2023.033375, 2022

[6] Daniel de Roux, Boris Perez, Andrés Moreno, Maria del Pilar Villamil, Cesar Figueroa, *Tax Fraud Detection for Under-Reporting Declarations Using an Unsupervised Machine Learning Approach.* DOI: 10.1145/3219819.3219878, 2018

[7] Abzetdin Z. Adamov, *Machine Learning and Advanced Analytics in Tax Fraud Detection*.DOI:10.1109/AICT47866.2019.8981758, 2019

[8] César Pérez López, María Jesús Delgado Rodríguez, and Sonia de Lucas Santos, *Tax Fraud Detection through Neural Networks: An Application Using a Sample of Personal Income Taxpayers.* DOI: 10.3390/fi110400862019

[9] Qinghua Zheng, Yiming Xu, Huixiang Liu, Bin Shi, Jiaxiang Wang, Bo Dong, *A Survey of Tax Risk Detection Using Data MiningTechniques*,DOI:10.1016/j.eng.2023.07.014,2023

[10] Beatrice Oyinkansola Adelakun, Ebere Ruth Onwubuariri, Gbenga Adeniyi Adeniran & Afari Ntiakoh, *Enhancing fraud detection in accounting through AI: Techniques and case studies*. DOI:10.51594/farj.v6i6.1232, 2023

[11] Adele Cutler, D. Richard Cutler and John R. Stevens, *Random forests.* DOI: 10.1007/978-1-4419-9326-7_5, 2011

[12] Jehad Ali, Rehanullah Khan, Nasir Ahmad, Imran Maqsood, *Random forests and Decision Trees*, year: 2012

[13] Lior Rokach and Oded Maimon, *Decision Trees*.DOI: 10.1007/0-387-25465-X_9,2005

[14] Soukaina Hakkal, Ayoub Ait Lahcen, *XGBoost To Enhance Learner Performance Prediction*.DOI:10.1016/j.caeai.2024.100254,2024

[15] Sakshi Indolia, Anil Kumar Goswami, S. P. Mishra b, Pooja Asopa, *Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach*. DOI: 10.1016/j.procs.2018.05.069,2018

[16] Sepp Hochreiter, Jürgen Schmidhuber, *Long ShorttermMemory*.DOI:10.1162/neco.1997.9.8.1735,1997

[17] Harsh H. Patel, Purvi Prajapati, *Study and Analysis of Decision Tree Based Classification Algorithms*, year: 2019. DOI:10.26438/ijcse/v6i10.7478.

[18] Keiron O'Shea1 and Ryan Nash, *An Introduction to Convolutional Neural Networks*, year:2015

[19] Benjamin Lindemann, Timo Müller, Hannes Vietz, Nasser Jazdi, Michael Weyrich, *A survey on long short-term memory networks for time series prediction.* DOI: 10.1016/j.procir.2021.03.088,2020

[20] Tianqi Chen, Carlos Guestrin, *XGBoost: A Scalable Tree Boosting System.* DOI: 10.1145/2939672.2939785