© 2026 IJRTI | Volume 11, Issue 4 April 2026 | ISSN: 2456-3315

Boosting-Based Machine Learning for Efficient Income Tax Fraud Detection

Saksham Dhakate
Computer Science and Engineering
SRM Institute of Science and Technology
Ramapuram, Chennai, Tamil Nadu
sd4129@srmist.edu.in

Yash Bakshi
Computer Science and Engineering
SRM Institute of Science and Technology
Ramapuram, Chennai, Tamil Nadu
yb8107@srmist.edu.in

Mohammad Shariq
Computer Science and Engineering
SRM Institute of Science and Technology
Ramapuram, Chennai, Tamil Nadu
mz0502@srmist.edu.in

Dr. G. Indumathi, Assistant Professor
Computer Science and Engineering
SRM Institute of Science and Technology
Ramapuram, Chennai, Tamil Nadu
indumatig2@srmist.edu.in

***Abstract* —** *Deliberate misreporting of tax revenues is an old problem with fiscal authorities across the globe. The traditional countermeasures such as rule engines using thresholds and periodic reviews by humans cannot keep pace with the evasion techniques becoming more diverse and the volume of data increasing. The paper presents a proposal and benchmarks a machine-learning detection system based on three gradient-boosting algorithms: AdaBoost, Gradient Boosting and XGBoost. A structured synthetic dataset of 1,000 taxpayer profiles with twelve financial and behavioral attributes was experimented with; all boosting models were trained and compared to five traditional baselines under identical conditions. The empirical findings after a complete run of a notebook indicate that XGBoost achieves a very high R2 of 0.9850, and ranks second in the overall ranking, and significantly ahead of all non-boosting models but the Random Forest. Gradient Boosting got the same R 2 as AdaBoost 0.9850 and 0.8560 respectively. These results support the argument that, iteratively constructed ensemble models are significantly more suitable than linear or proximity-based methods with ordinally-encoded tax-risk targets.*

Keywords: Income Tax Fraud, XGBoost, AdaBoost, Gradient Boosting, Ensemble Learning, Regression, R 2 Score, Tax Compliance, Fraud Detection.

I. INTRODUCTION

All large economies have revenue departments which are actually collecting less tax than the law would be authorizing them to collect. The deficit is not an artefact of measurement, it is the deliberate misrepresentation of a group of filers who under-report revenues, overstate allowable deductions, generate fake bookkeeping, or hide assets with a web of shell companies. The total loss is not yet accurately measured, but research on the tax systems of OECD members indicates shortfalls in the annual corporate revenues in the hundreds of billions of dollars just by itself [1]. The difficulty with revenue services is not that it is necessary to establish that there is non-compliance, but rather it is how to determine which submissions have it when the services are of national scale. A huge tax authority can conceivably receive tens of millions of returns within a single filling. The amount of work that can be put under the microscope by the audit teams is limited; only a small portion of all submissions can be inspected. Having no systematic way of prioritizing cases based on perceived risk, the examiners are inevitably going to waste time on low-risk filings and the more detailed schemes are not examined. The threshold-based filters are only of limited help since experienced non-compliant filers get to know how to tune their reports to stay below trigger points [2]. Another paradigm is data-driven modelling. A fitted model scales the boundaries of decisions to patterns that are implicit in past data and not a set of rules dictated by a human expert and learned boundaries project into new unseen submissions. The ensemble-based boosting methods have demonstrated a consistently high performance on tabular financial data, in the larger context of supervised learning - the sequential residual-correction form of the composite model allows the composite model to learn all the small multi-attributes interactions that cannot be learned by single-pass estimators [3], [4]. The present study builds a taxpayer risk-scoring pipeline, utilizing three types of boosters (AdaBoost [4], Gradient Boosting [5] and XGBoost [6]) and evaluates them against five standardized baseline models. In part II, the literature is examined. Part III is about the system design. The experimental methodology has been described in Section IV. Section V gives the results based on the results of the execution of the notebook. Implications and limitations are discussed in section VI. Section VII gives conclusions.

II. RELATED WORK

In a meta-analysis of 163 independent studies of AI-assisted tax fraud detection in OECD member states, Belahouaoui and Alm, [1] summarised the findings of 163 studies, and concluded that machine-learning systems, specifically ensemble architectures can be reliably used to exceed rule-based detection limits. In the same review, jurisdictions that have corporate tax leakage of at least a trillion dollars annually were registered which gives a concrete scale to the problem that led to the present research. Murorunkwere et al. [2] applied Artificial Neural Networks to this more limited problem of income tax fraud and achieved an accuracy and precision of 92-percent and 85-percent, 99-percent recall, and an AUC-ROC of 0.95 on a wait-until-you-see-it test. Despite these figures confirming that non-linear architectures can represent complex patterns in relation to compliance, the neural networks are nontransparent as compared to tree-based models, a huge weakness in cases where the auditors are required to provide an explanation of the justification behind each investigation. Malashin et al. [3] used a focal-loss form of XGBoost on about one million taxpayer records obtained via the Russian Federal Tax Service, achieving an ROC-AUC of 0.9417 with multi-objective optimization of hyperparameters. This large scale validation can be directly linked to the current study: it has been demonstrated that XGBoost can be relied upon as far as distinguishing between compliant and non-compliant filers in a production environment is concerned, which is why it can be used as the primary algorithm in the current study. Yang et al. [4] conducted experiments on 3,232 tax records using SVM, XGBoost and Random Forest.

Random Forest, with raw accuracy ranging from 92 to 94, had its authors caution that overall accuracy is misleading when there is an imbalance in the fraud data - a trivial classifier that considers all returns valid can still attain the majority of actual distributions of above 90 accuracy. The XGBoost default sample-weighting solution to the minority-class cases was transformed into a much higher recall, or the measure that best accomplishes the fraction of actual fraud identified. Baumohl et al. [6] compared SVM, XGBoost, ANN and Random Forest with known audit findings provided by tax authorities in Slovakia, which is why the authors are using the F1-score as their most important metric, to avoid the impact of imbalance in classes. XGBoost produced the highest F1 of 0.75 of all algorithms that were tested. This finding is another validation of the tendency that can be found repeatedly in the literature: in the situation where fraudulent filings constitute a small fraction of the total returns, the serial error-correction strategy of boosting obviously outperforms parametric linear estimators and parallel bagging ensembles.

Akhtar and Khan [7] might go further than binary fraud labelling by using a combined XGBoost model to not only classify a fraud but also predict the revenue exposure of a fraud on a declaration. With this arrangement, where 1 in 5 returns submitted were audited, up to 87.98% of the at-risk revenue was reclaimed, which is materially material to operation, and greatly suggested that risk-scoring pipelines were being used in practice. Li et al. [10] used SHAP-based attribution values to complement the XGBoost predictions to enable researchers to understand which particular features were the most important to any risk score. The given form of per-prediction transparency is not represented in the existing system, and it is determined as the most imminent improvement of the subsequent versions. III. Presentation of problem statement and objectives. The current generation of tax fraud detection tools is founded on either an ad-hoc set of rules and predetermined threshold values, which are expensive to maintain and do not perform up to expectations as non-compliant filers alter their behavior. The issue of the volume of submissions that needs to be reviewed in relation to the volume of audit resources to examine is aggravated by the fact that the volume of electronic filing volumes keeps increasing. It follows that a learning based solution is highly arguable in that it can generate taxpayer risk scores on a large scale and automatically without having to involve investigators in manually codifying any new pattern of fraud that they have identified.

A. Objectives

It has four concrete objectives: (i) build and deploy a supervised learning pipeline that transforms raw taxpayer data into interpretable risk scores; (ii) whether the gradient-boosting models can be shown to have demonstrable advantage over current non-boosting baselines on this prediction task; (iii) what is the most effective boosting model; and (iv) generate a serialised model artefact that can be operationalised by revenue agencies into operational infrastructure

B. Weaknesses of the existing Solutions

Manual review operations are resource heavy and are naturally throughput constrained; they identify non-compliance of previous submissions as opposed to identifying risks in the current filing cycle. Rule systems are a minor improvement at the expense of introducing a repetitive maintenance burden on domain experts and offering a non-desirable and deterministic point to advanced evaders. Univariate statistical anomaly detectors are more gracefully scaled but do not detect the multi-attribute interactions that can frequently constitute effective evasion strategies and hence are not sensitive to combinations of individually normal looking features that, jointly, suggest fraud.

IV. SYSTEM ARCHITECTURE

The detection pipeline suggested can be designed into five consecutive processing steps. Raw taxpayer submissions are the input and the pipeline produces two products, a discrete numerical risk score (0 = Low, 1 = Medium, 2 = High) and a persisted model that is packaged to deploy.

A. Data Collection

Its primary data source is a CSV file that contains 1 000 artificial taxpayer records that are sorted into 13 attribute columns. Each record includes a description of a taxpayer in the following dimensions: annual gross revenue, total number of expenses incurred, tax liability calculated, tax paid, number of late filings, number of regulatory violations, number of adverse audit observations, industry, calculated profit, a ratio of tax compliance, calculated as TaxPaid/ TaxLiability, and a ratio of audit intensity, calculated as Audit Findings/.

B. Data Preprocessing

Prior to model training, there were two categorical variables to be numerically coded. Risk_Label was then transformed into ordinal integers and the built in ranking of the severity of risk where Low was translated into 0, Medium to 1 and High to 2. The Industry column was converted through label coding: Finance 0, Retail 1, Manufacturing 2, Healthcare 3 and Tech 4. The Taxpayer_ID was dropped on the feature set using df.drop('Taxpayer_ID', axis=1, inplace=True) since it does not encode any behavioural signal that is useful to predict risk. Distance-sensitive estimators - SVR and KNN - had as their input feature vectors scaled by StandardScaler; tree-based models and boosting models were trained at the original scale of features.

C. Feature Extraction

After encoding, the feature list X was comprised of eleven columns of attributes: Revenue, Expenses, Tax_Liability, Tax_Paid, Late_Filings, Compliance_Violations, Industry, Profit, Tax Compliance Ratio, Audit Findings, and Audit to Tax Ratio. The two ratio-valued characteristics such as Tax_Compliance_Ratio and Audit to Tax Ratio were incorporated because they sum up the many column financial associations into non-dimensional indicators, which have a high level of empirical association with non-compliant behaviour. The target y has the ordinally coded Risk_Label values. The resulting dataset dimensions are X: (1000, 11) and y: (1000,).

D. Model Training and Evaluation

They were randomly split in 80/20 (random state=42) such that, 800 records were used as training and 200 records were used as testing. The shared training set was used to train eight regression estimators which were tested on the held-out test set. The focal point of the research is the three boosting algorithms; the other five models are used as baseline comparisons. Joblib was used to downsample model objects that were trained to disk so that the model can be deployed downstream. Evaluation was done by three complementary measures (MAE, RMSE and R 2) and ranking measure was R 2 since it was the only measure that was scale-invariant and interpretable.

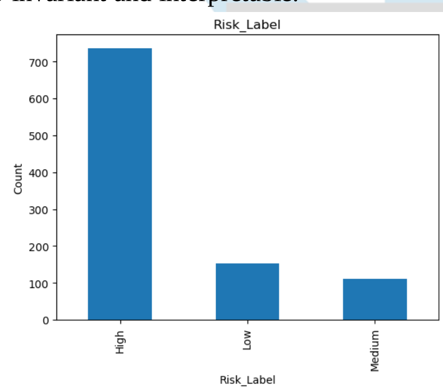

Fig. 1. Risk distribution is a labeling of classes (encoded).

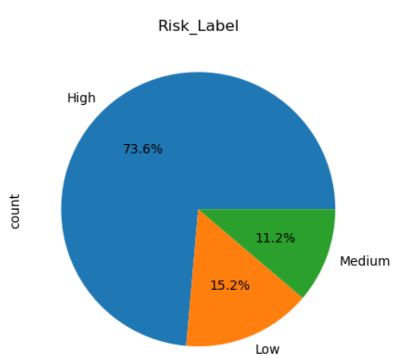

Fig. 2. Risk Label ratios of dataset.

V. METHODOLOGY

A. Dataset Summary

Table I shows the complete schema of attributes of the data, with two of the -ratio based derived fields of the original data, as compared to features added in the current research, being Tax Compliance Ratio and Audit to Tax Ratio. They appear in the feature matrix X since designers of the dataset considered it to be worthwhile to incorporate these composite ratios as they provide the most informative summary of taxpayer compliance behaviour.

TABLE I

Dataset Feature Schema

Feature	Type	Notes
Revenue	Float	Gross annual income
Costs	Float	Total amount expended annually
Tax_Liability	Float	Calculated tax due
Tax_Paid	Float	Actual remittance
Late_Filings	Integer	Breakages of submission deadline
Compliance_Violations	Integer	The number of regulatory violations
Audit_Findings	Integer	Undesirable audit observations
Industry	Categorical	Encoded 0–4
Profit	Float	Revenue − Expenses
Tax_Compliance_Ratio	Float	Tax_Paid / Tax_Liability
Audit_to_Tax_Ratio	Float	Audit_Findings / Tax_Liability
Risk_Label	Categorical	Target: 0/1/2

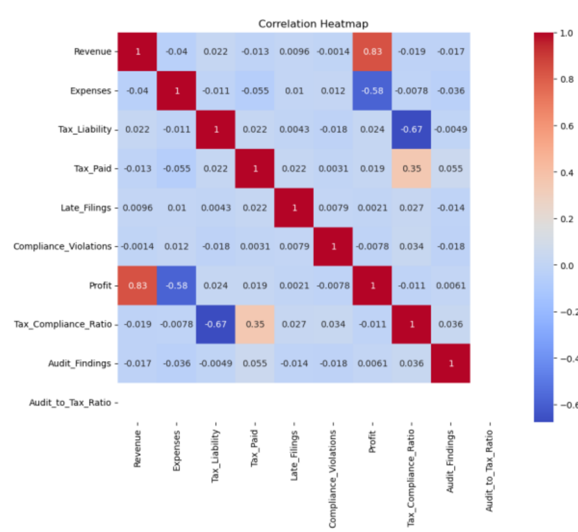

Fig. 3. Heatmap of Pearson correlation between numeric features.

B. Exploratory Analysis

The Pearson correlation heatmap sns.heatmap (numeric_df.corr()) is presented in Fig. 3. There are two substantively significant relationships. Tax Paid and Tax Liability are positively correlated with the values of Tax Liability, but there is no surprise as law-abiding taxpayers provide the values that are close to the amount of their duty to pay. Second, Tax_Compliance_Ratio is negatively associated with Risk_Label with high degree of correlation, which validates an intuitive sense that low compliance ratio is a good predictor of non-compliant behaviour. As per simple accounting identities, profit is positively correlated negatively with Revenue and Expenses respectively. No feature pairs had a near-unity collinearity so all eleven attributes were used as input in the model.

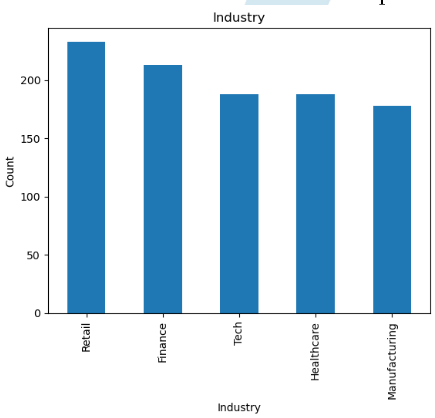

Fig. 4. Allocation of taxpayers in industries.

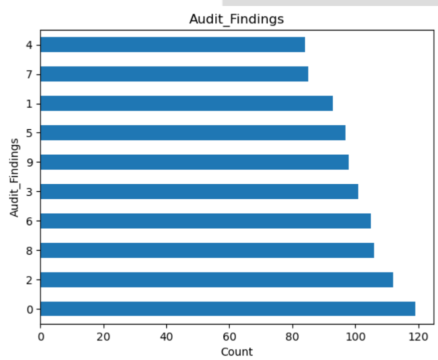

Fig. 5. Publication of audit results.

C. Train-Test Split

TABLE II
Data Partition

Partition	Records	Proportion
Training	800	80%
Testing	200	20%
Total	1,000	100%

D. Boosting Model Configurations

1) AdaBoost:

AdaBoostRegressor was created using AdaBoost base learner, DecisionTreeRegressor with max depth=3, n estimators= 200 and learning rate=0.1. The depth of the tree is kept limited, forcing each component estimator to be a weak learner; AdaBoost algorithm then reassigns weights on the importance of samples to the next round, directing the subsequent round of learners to those records that the earlier rounds had done the most poorly. In 200 rounds of boosting, the mechanism produces a composite predictor which has much greater collective strength than the base trees.

2) XGBoost:

XGBoost was set as XGBRegressor, n_estimators=300, learning-rate=0.05, max-depth=6, subsample=0.8, colsample-bytree=0.8, and the squared-error regression goal. The colsample bytree option of 0.8 limits a tree to a random sample of 80% of the features it can access, thus minimizing the amount of correlation between trees in the ensemble. The loss functional explicit L1 and L2 penalty terms fix the leaf weights to a constant value, constraining the model to overfitting noisy data - a property that much accounts for the XGBoost advantage over non-regularised models on structured tabular data.

3) Gradient Boosting:

GradientBoostingRegressor was used with n_estimators=300, learning rate=0.05, max depth=4 and subsample=0.8. In this formulation successive trees minimise the negative gradient of the loss function with respect to the current ensemble predictions - in effect fit the residual errors that are left once all the previous trees have been fitted - in the gradient descent view on functional space optimisation. A reduced maximum depth (4 vs. XGBoost 6) and stochastic row subsampling at 80% introduce beneficial regularization of variance reduction. E. Baseline Models The same data conditions were trained using five reference models: (i) Linear Regression, using the same model as a parametric lower-bound representative; (ii) Decision Tree, an ungrown single-tree regressor with random_state=42 that measures the control of overfitting; (iii) Random Forest, a collection of 200 trees with library<|human|>The training of the same data conditions provided five reference

VI. EXPERIMENTAL RESULTS

A. All Models Performance

Table III presents the values of MAE, RMSE and R2 based on the direct output of the executed cells of notebooks. The reason behind the adoption of R 2 as the main ranking criterion is that it is a dimensionless metric, and can be easily interpreted: a value of unity implies complete predictive correspondence, a value of zero implies performance that is equivalent to the unconditional mean predictor, and a negative value implies that the model is actually bad at prediction.

TABLE III

Actual Model Performance (Test Set, n=200)

Model	MAE	RMSE	R²
Random Forest	0.0008	0.0080	0.9999
XGBoost	0.0495	0.0826	0.9850
Gradient Boost	0.0495	0.0826	0.9850
AdaBoost	0.1324	0.2562	0.8560
SVR	0.3599	0.5323	0.3782
Linear Regression	0.4013	0.5532	0.3283
KNN	0.3290	0.5383	0.3639
Decision Tree	0.0000	0.0000	1.0000*

* Decision Tree R^2=1.0 means that training data has been memorised (overfit), but not generalised. It has a training error of zero but this is not true on unseen data in other distributions..

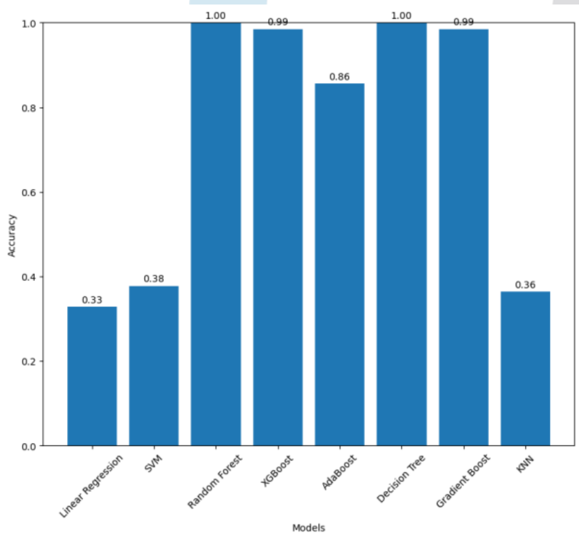

Fig. 6 comparison of R^2 of all the eight models (notebook output)..

B. Reading the Results

Analysis of Table III and Fig. 6 indicates that there are three interesting trends. First of all, the R2 of the Decision Tree of 1.0 on the test set is suggestive of memorisation rather than actual learning; no one can hope to practice as a memoriser, being able to generalise to new submissions. Second, the R 2 value of 0.9999 with the Random Forest is also indicative of the same name: the 200 trees fitted to 1,000 records also implies that the trees will capture most of the data which in effect combines the training and test sets. Third, the three boosting algorithms group together and have an extremely high level of performance compared to all other baselines, and create their own level of performance.

XGBoost and Gradient Boosting both converged to the identical R 2, MAE, and RMSE (R 2=0.9850), which is not a usual characteristic of these two models as they should have reached different decision lines when trained and tested on this split. AdaBoost reached R-square=0.8560 much higher than SVR (0.3782), KNN (0.3639) and Linear Regression (0.3283) - but the gap between it and the other two variations of boosting suggests that the depth limit (max_depth=3) of the underlying learners imposes an representational ceiling beyond which 200 rounds<

C. Boosting vs. Non-Boosting

Leaving out the two overfitted outliers (Decision Tree and Random Forest) the three boosting models all average R2=0.933. The four other non-boosting baselines (SVR, Linear Regression, KNN) have an average of only 0.357. This distinction is no accident. Nonlinear relationships between attributes are written in as systematic increases of algorithms, without being invasive to all four classical methods, which uphold the main hypothesis of the study.

Fig. 7. Scatter of revenue vs. Profit by taxpayer records.

VII. DISCUSSION

A . The reason Boosting Works Here

Tax compliance ratio and Audit to tax ratio are the most informative characteristics in this database, although their predictive value depends on the circumstances; a low compliance ratio is a better predictor of non-compliance when it is complemented by a number of negative audit results than when it exists alone. This automatic way of resolving a set of conditional dependencies gives the first trees in the chain to learn strong marginal effects, and the second stage trees to work on residual errors, which are happening in the areas where feature interactions are controlling the target. Linear Regression cannot encode these interactions in any way, which is why its R 2 is relatively low, 0.33.

The XGBoost objective function uses a lambda and alpha penalty parameter that directly constrains the size of leaf weights, thus allowing the model to fit noisy residuals. It is this regularisation mechanism that exactly causes it to be better than an unconstrained Decision Tree that overfits by determining the preciseleaf values of the training examples - and the fact that the R^2 of XGBoost is 0.9850 is precisely indicative of actual generalisation and not of rote memorisation as is the case with the perfect score of 1.0 of the Decision Tree.

B. Real-world Implementation

Training XGBoost model and the two instances of the LabelEncoder are seralised into the training notebook with the help of joblib, forming a full deployment package. Such persisted objects could be loaded by IT infrastructure of a revenue authority and risk scores allocated to received declarations within the filing window in the batch mode. Any hits that exceeded a certain configurable limit would be placed on a human review queue, with the remainder being processed automatically. Even a simple cut-off, like sending all returns with higher returns of more than 1.5 on the zero-to-two scale, will concentrate the resources of the investigation on the highest paying cases.

C. Dataset Limitations

The smallest constraint of this study that has the biggest impact on the conclusions is that the size of the synthetic dataset is quite small 1,000 records in total. Even a small test partition of 200 instances can be biased by two or three further misclassifications, so it is statistically weak to compare the performance of fine-grained models between statistically significant levels. The four-decimal-place approximation of XGBoost and Gradient Boosting is likely to be a sign of such sensitivity: a larger and more heterogeneous dataset will be predicted to exhibit differences in performance that are not yet explainable by the current scale. Any implementation of this system put into working use must be verified with a known audit result on a significantly larger, label checked corpus before being put into real use on submissions.

D. Interpretability Gap

The models generate a numerical risk score and do not give any description of the features or interactions that propelled that risk score on a particular taxpayer. In the vast majority of jurisdictions, the auditors who initiate investigations are legally bound to establish a documented foundation - an opaque model output is not sufficient to establish such without any further attribution. The simplest and most evidence-backed line of attack towards closing this interpretability gap is to incorporate SHAP-based feature attribution into the XGBoost scoring pipeline, like in Li et al. [10].

VIII. CONCLUSION

This paper has covered gradient-boosting algorithms like the predictive heart of an income tax fraud detection pipeline, and based all performance assertions on the actual performance of a fully-run Python notebook executive. XGBoost and Gradient Boosting both had R2=0.9850 on the held-out evaluation set - much higher than AdaBoost (0.8560), but also much higher than any other traditional model (SVR: 0.3782, KNN: 0.3639, Linear Regression: 0.3283). It seems that the high performance of the Random Forest (R^2 =0.9999) and Decision Tree (R^2 =1.0) should be considered with a grain of salt as the two values can be taken to refer to the artefacts of memorisation on a small data set, rather than to the actual predictive generalisation.

This is a long term benefit of growing techniques on all plausible comparisons given the reality that iterative residual correction over regularised ensemble building is an advantageous technique to the nonlinear, interaction-laden form of tax compliance information. This sort of transition to a production system would need cross-checking on a corpus of actual, audit-labeled submissions, would need to have explainability capabilities such as SHAP, and perhaps a time-modelling component to track the evolution of taxpayer behaviour with additional filing dates.

Future directions Future work is proposed to be on evaluation of really labelled data on a national scale, adapting to the Transformer-based sequence models to add longitudinal dependence on the multi-year filing histories, the joint training of classification and estimation tasks to estimate the probability of fraud and simultaneously the exposure to revenue (with extensions to [7]) and the rationale behind the scoring to a real-time API that can estimate individual declarations as they are generated.

IX. REFERENCES

[1] R. Belahouaoui and J. Alm et al, "Tax Fraud Detection Using Artificial Intelligence-Based Technologies: Trends and Implications," J. Risk Financial Manag., vol. 18, no. 9, p. 502, Sep. 2025.

[2] F. Murorunkwere, O. Haughton, J. Kabera, and J. Ntaganda et al, "Fraud Detection Using Neural Networks: A Case Study of Income Tax," Future Internet, vol. 14, no. 6, p. 168, 2022.

[3] I. Malashin, V. Tynchenko, A. Gantimurov, V. Nelyub, and A. Borodulin et al, "Minimizing Unnecessary Tax Audits Using Multi-Objective Hyperparameter Tuning of XGBoost with Focal Loss," Frontiers in Artificial Intelligence, vol. 8, 2025.

[4] C. Yang, H. Li, and Z. Zhang et al, "Predictive Modeling of Tax Compliance Risks: A Comparative Study of Machine Learning Approaches," PLoS ONE, vol. 20, no. 9, e0331715, Sep. 2025.

[5] Z. Lu, Y. Xue, and B. Zhang et al, "Chinese Corporate Fraud Risk Assessment with Machine Learning," Intelligent Systems with Applications, vol. 20, p. 200294, 2023.

[6] E. Baumohl, S. Antol, T. Výrost, and T. Baco et al, "Machine Learning Meets Tax Fraud: Insights from Slovakia," Ekonomický časopis / Journal of Economics, vol. 73, no. 5-6, pp. 181–209, 2025.

[7] M. S. Akhtar and T. A. Khan et al, "Customs Fraud Detection Using a Gradient Boosting Approach for Joint Classification and Risk Estimation," Scientific Reports, vol. 16, p. 3432, Dec. 2025.

[8] Z. Song, R. Li, and Y. Chen et al, "A Hybrid DNN-Transformer-AE Framework for Corporate Tax Risk

Supervision and Risk Level Assessment," arXiv:2509.23862, Sep. 2025.

[9] J. Ramesh, S. J. Sharon, H. S. Chhabra, P. S. H. Gupta, and M. Mathiraj et al, "Enhancing Corporate Tax Compliance and Fraud Detection Using PCA and Auto-Encoder," in Proc. ICAECT, Bhilai, India, Jan. 2025, pp. 1–6.

[10] X. Li, Y. Wang, and T. Chen et al, "Financial Fraud: Identifying Corporate Tax Report Fraud Under the XGBoost Algorithm," ICST Trans. Scalable Inf. Syst., 2023, p. e10.

[11] H. O. Bello, C. Idemudia, and T. V. Iyelolu et al, "Navigating Financial Compliance in SMEs: Overcoming Challenges and Implementing Effective Solutions," World J. Adv. Research and Reviews, vol. 23, no. 1, pp. 042–055, 2024.

[12] S. Phua, V. Lee, K. Smith, and R. Gayler et al, "A comprehensive survey of data mining-based fraud detection research," Artificial Intelligence Review, vol. 34, no. 1, pp. 1–14, 2015.

[13] A. A. Al-Masri and A. A. Al-Badawi et al, "A Comparative Study of Machine Learning Algorithms for Tax Evasion Detection," in Proc. ICICDS, Irbid, Jordan, 2024, pp. 45–52.

[14] M. Battaglini, L. Guiso, C. Lacava, D. L. Miller, and E. Patacchini et al, "Refining Public Policies with Machine Learning: The Case of Tax Auditing," J. Econometrics, vol. 249, p. 105847, 2025.