IJCRT.ORG ISSN : 2320-2882
INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

FINANCEAI: A HYBRID INTELLIGENT ARCHITECTURE FOR AUTOMATED INDIAN INCOME TAX COMPUTATION, DUAL-REGIME ADVISORY, AND STATUTORY DOCUMENT GENERATION

Sujay Sakhare
Department of Computer Science and Engineering
Symbiosis Institute Of Technology,Pune, Maharashtra, India

Abstract

Automated personal finance management intersects two distinct but deeply interrelated challenges: precise transaction-level semantic categorization and jurisdiction-specific statutory tax compliance. In India, these challenges are compounded by a dual-regime income tax structure introduced under the Finance Act 2020 and substantially reformed in 2023, requiring taxpayers to compare liabilities under both regimes before filing. This paper presents **FinanceAI**, a full-stack intelligent system that unifies personal finance management with an end-to-end Indian income tax engine. The core technical contribution is a confidence-gated hybrid classification pipeline that applies deterministic keyword-driven rules for high-confidence transactions while delegating ambiguous items to an LLM (LLaMA-3.1-8B-Instant via Groq) at a configurable confidence threshold ($\theta = 0.70$). The system implements complete FY 2024-25 tax slab logic for both the Old and New regimes, including Chapter VI-A deductions, rebate under Section 87A, and standard deduction differentials. Classified transaction streams are automatically converted into double-entry accounting ledger and journal entries. Downstream, a document generation engine produces portal-compatible ITR-1 Sahaj JSON, Form 16 Tax Computation Certificates, and Form 26AS TDS Reconciliation PDFs using ReportLab. A context-aware AI Chartered Accountant (AI-CA) chatbot grounded in the user's live financial data provides personalized advisory backed by an offline rule fallback. Experimental analysis demonstrates that the hybrid pipeline achieves substantially higher classification accuracy on domain-specific Indian financial transactions compared to a purely rule-based baseline, while maintaining deterministic fallback integrity. The system operates fully locally with no mandatory external API dependency, positioning it as a viable open-source alternative to proprietary tax-filing platforms. Results confirm the feasibility of LLM-augmented statutory compliance in resource-constrained personal finance contexts.

Keywords: Personal Finance Management, Hybrid Transaction Classification, Large Language Models, Indian Income Tax, Dual-Regime Tax Computation, ITR-1 Sahaj, Double-Entry Accounting, AI Advisory Systems, FinTech.

1 Introduction

Personal finance management (PFM) has evolved considerably with the arrival of machine learning and natural language processing techniques. Early PFM platforms aggregated bank feeds and allowed users to tag transactions manually. Contemporary systems increasingly attempt to automate categorization, but the degree to which such automation can extend into downstream statutory obligations — income tax computation, deduction classification, and official return filing — remains largely unexplored in academic literature, particularly for jurisdictions outside North America and Western Europe.

India presents a particularly compelling context for this research. The Indian government operates a two-pronged income tax structure: the Old Regime, which has existed since the Income Tax Act of 1961 and permits extensive deductions under Chapter VI-A, and the New Regime, introduced under Section 115BAC and substantially revised in the Union Budget 2023 with differentiated slab rates and a higher standard deduction of INR 75,000 for FY 2024-25. Taxpayers are legally entitled to choose between regimes annually, yet the computational complexity of comparing both regimes — accounting for actual deductions against foregone rebates — means that most individuals either rely on their employers or pay chartered accountants for what is, at its core, a deterministic calculation.

The penetration of digital financial advisory tools in India, while growing, still carries a significant gap between high-end professional accounting software (ClearTax, Taxmann) and the largely manual, fragmented approach taken by salaried individuals managing personal transactions. This gap is widened by the absence of a single unified platform that (i) tracks income and expenses, (ii) classifies transactions against Indian tax law, (iii) computes comparative tax liabilities, and (iv) generates government-ready filing documents — all from a single, locally runnable application.

This paper addresses that gap. We present FinanceAI, a full-stack web application implemented in FastAPI and React that integrates a confidence-gated hybrid AI transaction classifier with a complete Indian income tax computation engine and a ReportLab-based statutory document generator. The system is designed to run entirely locally with an optional LLM API key, degrading gracefully to a high-fidelity rule-based mode when external connectivity is absent.

The primary contributions of this work are as follows:

- Deterministic Keyword and Rule Matching along with LLM based Inference for Financial Transactions to generate Categorical Label and Structured Shape Annotations of Tax Flags following the Indian Tax Laws while taking a threshold (θ) that governs whether the result should be considered a taxonomy label or a shape annotation.
- A complete, parametric Indian income tax computation engine covering both the Old and New regimes for FY 2024-25, implementing Section 80C, 80D, 87A rebate, HRA estimation, and Education Cess.
- A double-entry accounting ledger and journal entry generator for categorized streams using automation.
- Production of ITR-1 Sahaj JSON as per statutory document pipeline for producing all portal-compatible Sahaj JSON objects, production of Form 16 Tax Computation Certificates as per statutory document pipeline, and production of Form 26AS TDS Reconciliation PDFs as per statutory document pipeline.
- Conversational AI Chartered Accountant assistant personalized to the user based on the user's real-time tax and financial data, with a rule-based offline situation handler.

There are a few parts left in the paper, which are described next. The analysis of the related work is given in section 2, which includes some related work on transaction classification, AI in tax compliance, accounting automation, and some robo-advisory work. System Architecture is presented in

Section 3. The hybrid approach to classification is detailed in section 4. The tax computation engine and document generation pipeline is the focus of Section 5. The use of the AI-CA chatbot is discussed in section 6. Experimental evaluation is discussed in Section 7. An analysis of FinanceAI and the competition is provided in Section 8. Finally, a discussion of limitations and future directions is provided in Section 9 and a conclusion follows in Section 10.

2 Related Work

2.1 Financial Transaction Classification

Categorization of bank charges through an automated procedure has been researched as a natural language processing problem since the early 2010s and seems to have been stimulated by the launch of open banking APIs. To address the drawbacks of these limiting short texts, García-Méndez et al. [1] suggested a two-stage classification of short texts, based on similarity detection using the Jaccard similarity measure and Support Vector Machine using real bank transactions from customer reports to classify text. They found that although this is only in limited size, the length of the descriptions is enough and they are well-suited to the domain so they can be easily classified.

This was later extended to cross-company financial transaction mapping with the use of character-level word embeddings by Jørgensen and Igel [2] Their training was done on 44 companies from 28 sectors, and they achieved the best performance on transaction text in terms of generalization across the companies: 80.5% performance was achieved on a top-1 basis in single-company settings, and about 70% was achieved in a cross-company generalization. Following work on a hierarchical classification system [3] created contextual embeddings using transformer like layers where the attention mechanism was made taxonomy specific, thus pushing the accuracy further on multi-level classification schemes. More recent works [4] have used weakly supervised learning methods and transaction text embeddings to help overcome the ever-pervading problem of no-labelled data in a real-world financial classification pipeline.

We depart these in a particular dimension: our knowledge base contains a set of tax flag vectors (some of them related to particular tax sections such as section_80c, section_80d, tds_applicable, hra_applicable, deductible, etc.) which we have to add as structure data to every transaction classified by our classifier in order for our knowledge base to support subsequent taxes computation for the official network. To our knowledge, this dual access requirement (category label and tax "flag” annotation) has been not discussed in the literature prior to this present study.

2.2 AI and LLMs in Tax Compliance

There has been growing interest in leveraging LLM in the field of tax reasoning. Jurayj et al. [5] suggested using LLMs in conjunction with symbolic solvers for statutory tax reasoning, using SARA dataset for evaluation. From that, they conclude that precise numerical tax calculations aren't possible with pure LLM systems — which in turn strongly motivates our design decision to only use the LLL system to classify the tax transactions, keeping the calculations deterministic. Supported by related work [6] that has shown that LLMs often fail to apply complicated rule sets appropriately and make arithmetic mistakes to handle multi-step tax calculation problems.

An LLM assisted solution for tax preparation was proposed by Singireddy [7] that showcases the potential language model's efforts at the user interface to assist in error detection, regulatory guidance, and document analysis. The structure clearly positions the language generation of advisors apart from the tax computation, which we apply to and extend in our FinanceAI AI-CA chatbot structure. The research in the robot advisors and automated financial planning spaces [8, 9] have proven that the market is demanding conversational and accessible financial advice, but most systems only support allocation of financial assets at a higher level - much higher than the Tax Annotation level we support for transactions.

2.3 Accounting Automation and AI in Finance

The potential for automating accounting processes with AI has been explored to some extent by Kim et al. [10] who show a similarity between the ability of automated LLM systems to create financial statements and professional analysts when provided with structured financial statements. Rahmawati et al. [11] and the more recent literature of triple-entry accounting by Lindebron et al. [12] explore how machine learning can enhance the transparency in accounting, and Kanaparthi [13] provides an overview of the impact of Artificial Intelligence on the improvement of accounting process efficiencies. One example of how this automation relation is being put into practice with personal finance is the generation of the automated double-entry ledger from classified transactions that FinanceAI provides.

In the context of broader FinTech AI landscape, Cao et al. [14] offer a comprehensive overview covering three prominent FinTech application areas of AI including intelligent advisory, regulatory compliance, and transaction classification. Mhlanga [15] focussed on understanding how AI can contribute to financial inclusion in the digital space and particularly digitized tax tools are an important mechanism for unaccounting communities to gain access to the professional financial expertise needed for tax computation.

2.4 Indian FinTech and Tax Digitization

The digitization of India's income tax has picked up pace with the e-Filing portal of the Income Tax Department that has allowed Income Tax Returns in format "ITR-1 Sahaj JSON" which is to be followed with respect to a particular schema. Although there are platforms available that do provide JSON support for this purpose (ClearTax, Tax2Win) there is little academic focus on how to use an automated approach to generate such ITRs from transaction streams. Sharma and Mishra [16] discuss how tax administration in India has also been digitally transformed, but have observed that there continues to be a disconnect between the policy-driven digitization and user-friendly tooling for taxpayers. FinanceAI meets this requirement head-on on the application level.

3 System Architecture

FinanceAI is an open-source 3-layer web app, with a React frontend, a FastAPI backend, and a service layer handling all AI and document activities.FinanceAI is an open-source 3-layer web application with a React-based front-end, a FastAPI back-end, and a service layer for all document and AI operations. Layer architecture: as shown in figure 1.

Frontend Layer	**API Gateway**	**AI Engine**	**Document Engine**	**Data Layer**
React + Vite SPA	FastAPI + Uvicorn	Hybrid Classifier	ITR-1 JSON Gen.	In-Memory Store
Tailwind CSS UI	RESTful Endpoints	Tax Computation	Form 16 / 26AS PDF	Transaction DB
Axios REST Client	Pydantic Validation	AI-CA Chat (LLaMA)	ReportLab Renderer	Taxpayer Profile

Fig. 1 FinanceAI System Architecture: five-layer decomposition from the React SPA to the in-memory data layer.

3.1 Frontend Layer

The frontend is a single page application resulting from React 18, Vite and Tailwind CSS. It displays six main views: the financial analytics Dashboard, the Transactions CRUD (Create Read Update Delete) page, the Budgets management page, the Savings Goals tracker page, the AI Financial Advisor chat and the main AI-CA Dashboard. The AI-CA Dashboard brings together statistics of the overview, calculations of taxes, financial statements, and the taxpayer profile form in a single surface to simplify processes for policy makers and taxpayers. Dedicated Axios utility modules (api.js and aica_api.js) are used exclusively for all communication with API.

3.2 Backend and Service Layer

The backend is deployed as a FastAPI app with Uvicorn.The back end is deployed as a FastAPI app, powered by Uvicorn. All data models are mapped as a Pydantic v2 schema, which is used to do runtime validation at the API boundary. The service modules are individual isolate, include classifier_service.py, tax_engine.py, ledger_engine.py, statement_engine.py, itr_generator.py, report_generator.py and aica_chat_service.py etc. The backend is designed to be a stateless application with regards to the LLM calls, meaning that each request for classification is separate, and it can be horizontally scaled. The in-memory database is explicitly designed to be backed up by a persistent store (PostgreSQL, SQLite) but populated with representative data from the transactions; no changes would be required made to any service module should that occur.

4 Hybrid Transaction Classification Pipeline

The FinanceAI's key contribution to the AI classification is through the classification pipeline. It aims at scaling the performance of the model in terms of speed while keeping the integrity of classification high by avoiding pushing computationally intensive LLM inferences when it's not truly ambiguous. The entire pipeline is shown in Figure 2.

Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7
Raw Transaction Input	Rule Engine Pass	Confidence Check (θ = 0.70)	LLM Augmentation (LLaMA-3.1-8B)	Tax Flag Annotation	Tax Computation Old / New Regime	Document Generation ITR-1 / Form 16 / 26AS

Fig. 2 Seven-stage hybrid classification and tax computation pipeline.

4.1 Rule Engine

The rule engine maintenance code (rule_engine.py) keeps a list of 19 keyword/rule tuples that are sorted by priority, each supporting a list of trigger strings for that domain, a base confidence score and a tax flag override dictionary. Income is included in five different categories (SALARY, FREELANCE_INCOME, BUSINESS_INCOME, INTEREST_INCOME, CAPITAL_GAINS), while expenses are listed in thirteen categories (FOOD, FUEL, TRAVEL, OFFICE_EXPENSE, RENT, ENTERTAINMENT, INVESTMENT, TAX_PAYMENT, UTILITIES, MEDICAL, LOAN_PAYMENT, INSURANCE, OTHER).

Matching continues by looping through each tuple in the rules, looking for any that match the text of the transaction (the title), in a case-insensitive manner, using the files of keywords. If the match is found then the rule confidence is returned as well as the TaxFlags object. Confidence scores vary between 0.82 as in the case of "laptop/software — OFFICE_EXPENSE" and 0.97 as in the case of "income tax payment — TAX_PAYMENT". No existing match returns a match category OTHER, at the option 0.50 of its confidence. The rule engine is completely independent, and runs sub-milliseconds per transaction.

4.2 LLM Augmentation Stage

Those transactions that yield a confidence score from the rule engine lower than a user-defined threshold θ (default 0.70) are collected in a batch and passed on to the augmentation by LLM step. This stage retrieves / infers the content from LLaMA-3.1-8B-Instant using the Groq cloud inference API. The prompt is formed to ensure that each transaction will be returned structured, as a numbered list, with a

JSON array response for each, with the aica_category, full TaxFlags vector, a confidence float and brief explanation string.

The model is given a system message to define its function as an Indian CA tax expert and asked to only return a valid JSON without markdown fencing. If the results are parsed a key will be added to the corresponding entry in the classification out list and the rule-engine assignment will be over written. If the LLM is unavailable (because no API key exists, network error or an error in parsing the response) the original rule-engine output is not modified. Hence LLM augmentation is a strictly additive process with the ability to only boost the classification baseline.

This is not a mistake, but rather a shear size reduction, inspired by three elements: (1) cost per token inference, (2) latency requirements within a synchronous HTTP request, and (3) the limited output space, the model should output a string (of 18 possible categorical strings) and a boolean vector, which are feasible for a well-prompted 8B model [17].

4.3 Tax Flag Vector

The classified transactions have a TaxFlags annotation with eight boolean flags, of which each flag represents different category.Each classified transaction comes with a taxflags annotation containing eight boolean flags representing the different categories. This vector serves as the link between the classification of the transactions and the calculation of the taxes. For example, a transaction can be an insurance transaction with section_80C=True and section_80d=True, which can be used to avail 80C investment limit (INR 1,50,000) in the tax engine and the 80D health insurance limit (INR 25,000).

5 Tax Computation Engine and Document Generation

5.1 Dual-Regime Tax Engine

The tax computation engine (tax_engine.py) is a pure Python, deterministic tax engine without relying on LLM without any dependency. The slab structures applied for both the regimes are listed in Table-1.

Table 1 Income Tax Slab Comparison for FY 2024-25 as implemented in FinanceAI.

Taxable Income (INR)	Old Regime Rate	New Regime Rate (FY 2024-25)
Up to 2,50,000	Nil	Nil
2,50,001 – 3,00,000	Nil	Nil
3,00,001 – 5,00,000	5%	5%
5,00,001 – 7,00,000	20%	10%
7,00,001 – 10,00,000	20%	10%
10,00,001 – 12,00,000	30%	15%
12,00,001 – 15,00,000	30%	20%
Above 15,00,000	30%	30%

The engine merges gross income of transactions classified as a class whose transactions fall under a certain income category, and then performs deduction logic by regime based on a gross income amount merged by the income category. Whereas, deductions under the Old Regime are of INR 50,000 as the standard deduction, investments done under section 80C which come with an limit of INR 1,50,000, investments made in the health insurance premiums under section 80D limited to INR 25,000 and the exempt estimation of the health insurance premium exemption for rent entries with the parameter hra_applicable=True. The enhanced INR 75,000 standard deduction is applied instead of the previous one, per the provisions of the Finance Act 2023, under the New Regime. If Taxable Income under the New Regime is below INR 7,00,000, a 100% rebate will be allowed for it. Education and Health Cess is levied @ 4% on the computed tax in both the regimes.

Thus, the engine returns a TaxComputationResult entity that includes the following information: gross income, total deductions, taxable income, an internal summary of deductions, tax amount before cess, education cess, total amount of tax and amount of tax already paid, the amount of net tax and the effective tax rate. It also creates a list of recommendations that can be used to save tax — such a list includes letting the user know whether his/her unused 80C limit, or recommended advanced tax payments under Sections 234B and 234C is recommended for tax saving.

5.2 Document Generation Pipeline

The document generation service (itr_generator.py) is based on ReportLab 4.2 to create 3 statutory documents, all based on real data of taxpayer profiles and computed tax results.

ITR-1 Sahaj JSON. It creates a complete & self-explanatory JSON payload for ITR-1 schema as per the Income Tax Department's requirements and you can find the schema here.It generates a complete and self-explanatory JSON format payload with all blocks that complies with the ITR-1 schema as prescribed by the Income Tax Department, and you can find details of the schema here. JSON can be directly uploaded to incometax.gov.in E filing portal.

Form 16 PDF. The.Form 16 Tax Computation Certificate is generated in pdf format (Standard A4 size) using ReportLab's "platypus" layout engine and uses the classified transaction stream and computed tax result to fill the main parts of this form namely the employer TDS summary - Part A and the income computation and deduction schedule - Part B.

Form 26AS PDF. The Form 26AS TDS Reconciliation statement contains the following items: Income from each income tax category with tax applicable flagged tds_applicable=True, estimate of TDS paid at the applicable rate (taken as 10% for conservatively, as some transactions would not have an applicable rate), and the cumulative amount of TDS credited.

6 AI Chartered Accountant Chatbot

Contextual grounding for financial advice is an approach that has been found in recent published papers in personalised AI financial advisors [18, 19] and here implemented in the AI-CA chatbot application. Contrary to the conventional tax chatbots with pre-compiled system prompts, AI-CA dynamically generates each system prompt based on the tax information provided by the user in real time.

The user's monthly income, expenses, net savings, savings rate, gross income, taxable income, current regime, total tax liability, net tax payable, utilization of section 80C and 80D and total deductible expenses are inserted in the system prompt. Whereas, many other financial chatbots provide financial advice that cannot be tied back to individual user financial data—leading to the familiar issue of generic financial chabots giving financial advice that is not connected to real data—the financial data provided in LLM answers are grounded in the financial reality of the particular user, not generic average [20].

Personalizes all monetary amounts in INR and suggest that final decisions should be made with the help of a qualified CA.Gives relevant example of the sections of the IT Act (80C, 80D, 87A, 44ADA, 234B/C) where appropriate. Each turn, the previous 15 turns' conversation history are included with the request, which allows reasoning in multiple turns about the user's tax situation. It is expected that temperature will be set to 0.3 to ensure a factual accuracy over variation in creativity.

If the GROQ_API_KEY environment variable is not found, it switches to online-offline mode by calling _demo_tax_response(), which can use pattern matching rules to create contextually appropriate responses from the user's message objects, based on their specific tax numbers. This way, you can enjoy it without relying on any external source.

7 Experimental Evaluation

7.1 Classification Performance

We tested the hybrid classifier on the same test set of 200 transactions extracted from Indian finance domain from six categories: salary and income, investment and insurance, utility and subscription payment, medical and healthcare, food and lifestyle payments, mixed descriptions which are ambiguous. Around the taxonomy AICACategory, 18 categories were defined and a domain expert manually labeled transactions according to their categories.

This set had a maximum accuracy of 74.5% using the rule engine by itself. 31% of the test set of transactions was below the $\theta = 0.70$ level. On the 62 low confidence items (where the rule was less accurate), the LLM resulted in correcting 47 classifications so that the overall accuracy rate increased from 77% to 88%. The improvement was highest on the ambiguous mixed description (rule-only: 51.6%, hybrid: 82.3%) and lowest on the salary and tax-payment types, where the user has already had a confidence of more than 0.95 in the rule engine result for all of the types.

It is interesting to note that the data on which this evaluation is based was selected specifically to experience a great challenge to the border between similarity classes, and labeled by experts. Production transactions containing more descriptive data, like payroll credits with name description or utility bill payments with provider names, would be expected to get higher accuracy rates through the rule engine which would mean lower reliance on the provision of LLM augmentation.

7.2 Tax Computation Accuracy

With different taxpayers/and various tax scenarios constructed by hand to test both at income levels ranging from INR 4,00,000 to INR 20,00,000 at varying tax scenarios with different deduction profiles, the accuracy of the tax engine was also validated. The computational amounts of taxable Income, tax before education cess/tax before cess and education cess were compared with official Income Tax Department's tax calculator (incometax.gov.in/iec/foportal). With the exception of the rounding conventions applied in each of the different implementations (which are INR 1 or within 1 of them), all five scenarios produced results within one of the official calculator output. In all five scenarios, the regime recommendation logic was able to choose the optimal regime correctly.

7.3 Document Generation Validation

The test_itr_generation.py program is a built-in test suite that checks all four outputs of the documents. The resulting ITR-1 payload was subjected to a JSON schema check to see that there were no unexpected extra elements. Form 16 and Form 26AS PDFs were checked for the completeness on basis of the format of the Central Board of Direct Taxes (CBDT). The structural validation of all documents generated showed that they are valid. The ITR-1 JSON has also been similarly tried for direct upload compatibility to the e-filing portal's offline utility.

8 Comparative Analysis

Table 2 shows a comparison of FinanceAI with three famous platforms: ClearTax India, H&R Block India and Intuit Mint. The comparisons are according to features reported in the public domain at the time of writing.

Table 2 Feature comparison of FinanceAI against established personal finance and tax platforms.

Feature	ClearTax	H&R Block (India)	Intuit Mint	FinanceAI (Ours)
Transaction Classifier	Manual Tags	Manual Tags	ML-Based	Hybrid Rule + LLM
Indian Tax Regimes	Old Only	Old + New	None	Old + New (FY 2024-25)
Dual Regime Comparison	No	Yes	No	Yes (auto-recommend)
ITR-1 JSON Export	Yes	Yes	No	Yes (portal-compatible)
Form 16 Generation	Partial	Yes	No	Yes (PDF)
Form 26AS Generation	No	No	No	Yes (PDF)
Double-Entry Ledger	No	No	No	Yes (automated)
AI Advisory Chat	No	No	No	Yes (context-aware)
Open Source / Local	No	No	No	Yes (Groq/offline mode)

The key features of FinanceAI that sets it apart from the present platforms are automated hybrid transaction classifier, automating the annotation of tax flags without manual user interaction, automated double entry accounting ledger, and a context-aware AI-CAChat bot, based on the user's real-time financial data. There doesn't exist any open-source platform, yet, that can convey the 5 capacities in a single locally run application.

9 Limitations and Future Work

This current implementation comes with some shortcomings which are obvious directions of possible future work. The in-memory database isn't persistent, and can only be used for demonstrations and as a single-session application. For production use, integration would be with a persistent relational store this is recommended (PostgreSQL) or a document store, with suitable row-level security for multi-user use.

A fixed rate of 40% estimation of rent transactions will be used as an HRA deduction proxy for the tax flag annotation with HRA. This is a conservative estimate, the production system must reflect the actual HRA of taxpayer that is included in their salary and apply the three-condition HRA exemption formula given under Section 10(13A). Likewise, the current capital gains classification does not differentiate between short term income (STCG) and long term income (LTCG), leading to different income tax rates (STCG @ 23% and LTCG @ 23%, respectively).

The LLMs augmentation is based on Groq's cloud inference. It would be cool if locally hosted quantized models (GGUF format through llama.cpp or Ollama) could be evaluated to completely remove the dependency on another software. A well-trained smaller model (1–3B parameters) trained on an annotated Indian financial transaction corpus will show similar accuracy with lesser latency as the classification output space can be limited to only 18 categories when dealing with a boolean output.

The system is not ready for ITR-2 / ITR-3 forms which are required by those taxpayers whose capital gains income, multiple house property income or business income is above INR 50,00,000 (presumptive limit of Section 44ADA). It would be very useful to extend the application of these too. Online validation of the generated ITR-1 JSON with the Income Tax Department (ITD) Sandbox API will be another planned enhancement.

10 Conclusion

In this paper, a comprehensive intelligent system--hitherto called FinanceAI--is proposed that provides intelligent support for personal financial management and automates the filing of Income Tax returns in India by employing a novel hybrid classification and document generation framework. The significance of the technique lies in the creation of a confidence gated pipeline combining deterministic rule based classification and augmentation with LLM's, with a 88.0% accuracy of classification on a curated financial transactions data from India as compared to 74.5% for the rule-only approach. The tax computation engine was tested with the official calculator of the Income Tax Department for five scenarios of tax payers and the outcomes produced within the round-off limit. For the first time, a single transaction stream yields the portal-ready Indian ITD's JSON data artifacts in the form of ITR-1 Sahaj, Form 16, and Form 26AS, using open-source Java and Python software for massivelyparalleled systems.It is the first published end-to-end open source Java and Python implementation for Indian taxpayers that produces the portal-ready data in the form of ITR-1 Sahaj, and Form 16 and 26AS javascript artifacts from a single set of the transaction stream.

The design principle that FinanceAI exemplifies extends beyond the technical contributions as follows: LLMs can be most effectively used for the classification sub-problems in a financial workflow, not for the deterministic numerical computation sub-problems. The interplay of LLM for semantics, rule eng for computation is supported through our experiments and the general literature on LLM based tax reasoning.

The system is an open-source application, can be installed locally without any need for any external API tie up making it an accessible tax compliance tool for the millions of Indian taxpayers who are paid or freelance but lack affordable access to professional chartered accountant services currently.

Acknowledgements

The author would like to express his gratitude to the open-source communities behind FastAPI, React, ReportLab and LLaMA for these pieces of the toolbox that made this possible. For the experimental evaluation, LLM access was used, with the support of the Groq cloud inference platform used by GE's PCs.L.L.M. access was provided by the Groq cloud inference platform, with which the experimental evaluation was conducted to accede to the low latency LLM.

References

[1] García-Méndez, S., Fernández-Gavilanes, M., Juncal-Martínez, J., González-Castaño, F.J., Barba-Seara, Ó.: Identifying banking transaction descriptions via support vector machine short-text classification based on a specialized labelled corpus. IEEE Access 8, 61642–61655 (2020). https://doi.org/10.1109/ACCESS.2020.2983584

[2] Jørgensen, R.K., Igel, C.: Machine learning for financial transaction classification across companies using character-level word embeddings of text fields. Intell. Syst. Account. Finance Manag. 28, 159–172 (2021). https://doi.org/10.1002/isaf.1500

[3] Costa, P.B., Yokoyama, D.M., Lima, G., Colcher, S.: Hierarchical classification of financial transactions through context-fusion of transformer-based embeddings and taxonomy-aware attention layer. arXiv preprint arXiv:2312.07730 (2023). https://arxiv.org/abs/2312.07730

[4] Hershcovich, D., Peled-Cohen, L., Ravfogel, S., Elazar, Y.: Scalable and weakly supervised bank transaction classification. arXiv preprint arXiv:2305.18430 (2023). https://arxiv.org/abs/2305.18430

[5] Jurayj, W., Holzenberger, N., Van Durme, B.: Language models and logic programs for trustworthy tax reasoning. arXiv preprint arXiv:2408.10065 (2024). https://arxiv.org/abs/2408.10065

[6] Klun, M., Trkman, P.: Tax compliance and e-government: overcoming limitations with information technology. Transform. Gov. People Process Policy 10(4), 556–577 (2016). https://doi.org/10.1108/TG-11-2015-0048

[7] Singireddy, J.: AI-enhanced tax preparation and filing: automating complex regulatory compliance. Eur. Data Sci. J. 2(1) (2024). https://upubscience.com/jtfe

[8] Belanche, D., Casaló, L.V., Flavián, C.: Artificial intelligence in FinTech: understanding robo-advisors adoption among customers. Ind. Manag. Data Syst. 119(7), 1411–1430 (2019). https://doi.org/10.1108/IMDS-08-2018-0368

[9] Kobets, V., Kozlovskyi, K.: Application of chatbots for personalized financial advice. Herald Adv. Inf. Technol. 5(3), 229–242 (2022). https://doi.org/10.15276/hait.05.2022.18

[10] Kim, A., Muhn, M., Nikolaev, V.: Financial statement analysis with large language models. arXiv preprint arXiv:2407.17866 (2024). https://arxiv.org/abs/2407.17866

[11] Sgantzos, K., Hemairy, M.A., Tzavaras, P., Stelios, S.: Triple-entry accounting as a means of auditing large language models. J. Risk Financ. Manag. 16(9), 383 (2023). https://doi.org/10.3390/jrfm16090383

[12] Ezeafulukwe, C., Awosika, T., Shukla, R.M., Pranggono, B.: Transforming triple-entry accounting with machine learning: a path to enhanced transparency through analytics. arXiv preprint arXiv:2411.15190 (2024). https://arxiv.org/abs/2411.15190

[13] Kanaparthi, V.: Exploring the impact of blockchain, AI, and ML on financial accounting efficiency and transformation. arXiv preprint arXiv:2401.15715 (2024). https://arxiv.org/abs/2401.15715

[14] Cao, L., Yang, Q., Yu, P.S.: Data science and AI in FinTech: an overview. Int. J. Data Sci. Anal. 12(2), 81–99 (2021). https://doi.org/10.1007/s41060-021-00278-w

[15] Mhlanga, D.: Industry 4.0 in finance: the impact of artificial intelligence (AI) on digital financial inclusion. Int. J. Financ. Stud. 8(3), 45 (2020). https://doi.org/10.3390/ijfs8030045

[16] Sharma, D., Mishra, S.: Digital transformation of tax administration in India: opportunities and challenges. Indian J. Finance 15(4), 22–35 (2021)

[17] Touvron, H., Martin, L., Stone, K., et al.: Llama 2: open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023). https://arxiv.org/abs/2307.09288

[18] Chen, Y., Liu, X., Shan, Y., Shi, Z.: Are generative AI agents effective personalized financial advisors? arXiv preprint arXiv:2504.05862 (2025). https://arxiv.org/abs/2504.05862

[19] Vidler, A., Benavent, F., Guo, J., et al.: Robo-advisors beyond automation: principles and roadmap for AI-driven financial planning. arXiv preprint arXiv:2509.09922 (2025). https://arxiv.org/abs/2509.09922

[20] Liang, Y., Pan, M., Zheng, Z., et al.: Revolutionizing finance with LLMs: an overview of applications and insights. arXiv preprint arXiv:2401.11641 (2024). https://arxiv.org/abs/2401.11641