iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** XI **Month of publication:** November 2026

DOI: https://doi.org/10.22214/ijraset.2026.85886

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue XI Nov 2026- Available at www.ijraset.com

FinCoach: AI-Powered Personal Finance Coach for Intelligent Expense Management and Personalized Financial Planning

Mr. Sarthak Ranade[1], Prof. Vidya B. Kale[2], Ms. Rajashri Chavan[3], Ms. Tejal Avhad[4], Mr. Krushna Kadam[5]
Dept. of Computer Engineering, Sir Visvesvaraya Institute of Technology, Nashik, India

Abstract: ***Managing personal finances remains difficult for many individuals because daily transactions are tracked manually, categorized inconsistently and rarely analyzed for patterns. This paper presents the design of an AI-powered personal finance coach: a web-based system that ingests bank statements in CSV format or manual entries, cleans the data, automatically categorizes each transaction, analyzes income, expenses, savings and cash flow, flags unusual spending, forecasts future expenses, and generates personalized budget and saving suggestions. Results are delivered through interactive dashboards and a conversational assistant that explains the findings in simple language. A preliminary offline evaluation of the AI core on a 2,400-transaction synthetic dataset shows that a TF-IDF and logistic regression categorizer reaches 92.29% accuracy over five categories, that an Isolation Forest detector attains an F1-score of 0.925 against 0.689 for a per-category z-score baseline, and that a seasonality-aware regression forecaster achieves a MAPE of 4.19% on held-out months. These results demonstrate the feasibility of the pipeline; validation on real user data is identified as future work.***

Index Terms: ***personal finance management, expense categorization, anomaly detection, expense forecasting, machine learning, chatbot, budget planning.***

I. INTRODUCTION

Personal finance management has become more complex as digital payments, UPI transfers, card transactions and subscription services generate a continuous stream of small transactions. Most individuals do not have the time or the tools to review these records regularly, so they lack a clear picture of where their money goes, struggle to build realistic monthly budgets and overlook unusual or unnecessary spending. Manually recording and categorizing every transaction is time-consuming and error-prone, and spotting an irregular expense in a long statement is even harder.

Recent advances in machine learning (ML) and large language models (LLMs) make it possible to automate these tasks. Text classification can assign a category to a transaction from its description, unsupervised anomaly detection can highlight outliers without labelled examples, time-series and regression models can estimate future expenses, and conversational assistants can explain numerical results in plain language. Several recent studies combine subsets of these capabilities [1]–[3], but most address expense tracking, budgeting or chat in isolation, and many list bank-statement upload, stronger security and better personalization as open items.

This paper proposes an AI-powered personal finance coach that integrates these functions in a single web-based platform. The main contributions are: (i) an end-to-end architecture covering CSV ingestion, data cleaning, categorization, spending analysis, anomaly detection, forecasting, budget recommendation, dashboards and a chatbot; (ii) a modular processing pipeline in which each AI component can be replaced independently; and (iii) a preliminary evaluation of the categorization, anomaly detection and forecasting components on a controlled synthetic dataset, including comparison with simple baselines.

The remainder of this paper is organized as follows. Section II presents the literature survey. Section III states the problem and objectives. Section IV describes the proposed system. Section V covers implementation and requirements. Section VI reports experimental results. Section VII discusses limitations, and Section VIII concludes the paper.

II. LITERATURE SURVEY

Chaugule et al. [1] describe an AI-driven system for personal finance and investment recommendation built with React.js, Tailwind CSS and PostgreSQL, using ML-based expense categorization and AI recommendations; the authors list banking APIs, multiple currencies, a mobile application, advanced forecasting and improved personalization as future work.

©IJRASET: All Rights are Reserved

Divya et al. [2] combine expense tracking with a chatbot in a Django and MySQL application, using LayoutLM for receipt processing and supporting categorization, anomaly detection and expense forecasting; real-time bank or UPI integration, multilingual support and stronger security remain open. Manasa et al. [3] use machine learning to track expenses and an LLM to predict future spending, with budget management and automatic alerts, and note the need for better security, advanced prediction models, bank-statement upload and mobile support. Table I summarizes these studies.

Other work covers neighbouring problems. Expense-tracker web applications are studied in [4] and a real-time expense tracker using ML in [5]. Agarwal et al. [6] and Talasila [7] discuss AI assistants for financial literacy, budgeting and planning. Smart Expense [8] adds a CNN-based component and anomaly detection to a finance tracker, while Kulshrestha [9] addresses real-time tracking and forecasting. Smart expense tracking with ML is considered in [10], OCR techniques for financial documents in [11], and a review of AI-driven finance assistants for budgeting and expense automation in [12]; an AI-powered finance platform is presented in [13].

These works show that individual capabilities are well explored, but the surveyed papers report that statement-based ingestion, security, forecasting quality and personalization are still evolving. Our work targets this gap by combining CSV-statement analysis, categorization, anomaly detection, forecasting, recommendations and a conversational explanation layer in one modular design.

TABLE I
Summary of Surveyed Personal Finance Systems

Ref.	Year	Methodology	Key Findings	Limitations / Future Work
[1]	2026	Agile SDLC; React.js, Tailwind CSS, PostgreSQL; ML-based expense categorization; AI recommendations	Automates expense tracking, budgeting and investment planning; budget alerts and personalized recommendations	Banking APIs, multiple currencies, mobile app, advanced forecasting, better personalization
[2]	2026	Django and MySQL web application; chatbot; LayoutLM for receipt processing	Combines expense tracking, chatbot, categorization, anomaly detection and forecasting	Real-time bank/UPI integration, multilingual support, stronger security, more personalization
[3]	2025	Machine learning for expense tracking; LLM for expense prediction	Expense tracking, prediction, budget management and automatic alerts	Better security, advanced prediction models, bank-statement upload, mobile app

III. PROBLEM STATEMENT AND OBJECTIVES

A. Problem Definition

Five practical problems motivate this work. First, daily expenses are difficult to track regularly. Second, users often lack insight into where and how their money is spent. Third, manual recording and categorization is slow and introduces errors. Fourth, unusual or unnecessary expenses are hard to identify in large transaction records. Fifth, users struggle to build a realistic monthly budget from their actual spending habits.

B. Technical Challenges

Building a system that addresses these problems raises five technical challenges: (1) collecting and processing large numbers of transactions from CSV bank statements; (2) cleaning missing, duplicate, inconsistent or incorrect records before analysis; (3) automatically assigning categories such as Food, Shopping, Travel, Bills and Entertainment from noisy free-text descriptions; (4) detecting unusual or excessive spending patterns in a user's history; and (5) estimating future expenses from historical data to support budget planning.

C. Objectives

The objectives are to (a) automate expense management by categorizing transactions and analyzing income, expenses, cash flow and trends; (b) detect unusual spending; (c) predict future expenses from historical data; (d) generate personalized budgets and saving suggestions; and (e) present insights through interactive dashboards and an AI chatbot that explains financial information in simple language.

IV. PROPOSED SYSTEM

A. System Architecture

The proposed system follows a four-layer architecture (Fig. 1). The presentation layer provides a web interface, an interactive dashboard and a chatbot window. The application layer, implemented with the Django framework, handles authentication, CSV upload and manual entry, and the budget and recommendation logic. The AI/ML layer contains data cleaning, the transaction categorizer, the anomaly detector, the expense forecaster and an LLM-based assistant accessed through an API. The data layer stores user and transaction records in a relational database (MySQL or PostgreSQL), together with trained models and the uploaded raw statements.

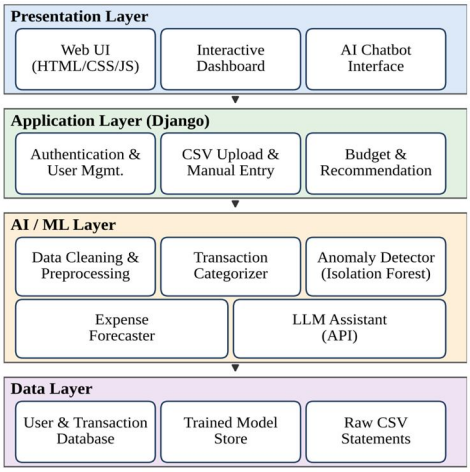


Fig. 1. Layered architecture of the proposed AI-powered personal finance coach.

B. Processing Workflow

Fig. 2 shows the end-to-end workflow. The user uploads a CSV statement or enters transactions manually. The data are validated and cleaned; features are extracted from the description, amount and date; each transaction is categorized; and the system computes spending statistics, detects anomalies and forecasts the coming month. Finally, budget and saving suggestions are generated and shown on the dashboard, where the chatbot can explain them.

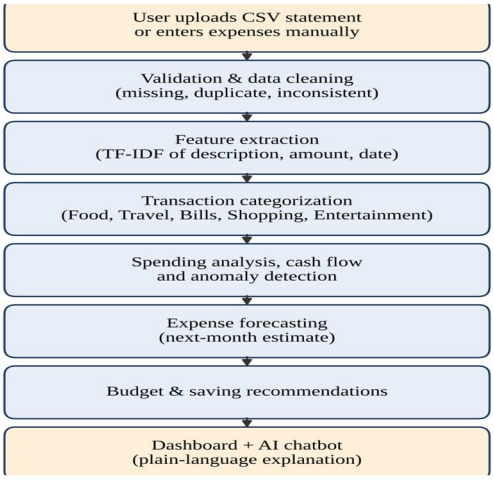


Fig. 2. Workflow of the proposed system from data input to dashboard and chatbot.

©IJRASET: All Rights are Reserved

C. Data Input and Cleaning

Users may upload a CSV bank statement or add transactions manually. The cleaning module validates the schema (date, description, debit/credit amount), parses dates, removes exact duplicate rows, handles missing or malformed amounts, normalizes description text (lower-casing and removing reference numbers and punctuation) and flags inconsistent records for review.

D. Transaction Categorization

Each cleaned description is converted to a TF-IDF vector over word unigrams and bigrams [16]. For a term t in description d drawn from a corpus of N descriptions,

$$w(t,d) = tf(t,d) \cdot \log(N / df(t)) \quad (1)$$

where tf is the term frequency and df the number of descriptions containing t. A multiclass classifier then predicts one of the categories Food, Shopping, Travel, Bills and Entertainment. Logistic regression is used as the default because it also outputs class probabilities, so low-confidence predictions can be sent to the user for confirmation, and user corrections can be fed back as new training examples.

E. Spending Analysis

From the categorized data the system computes total income I, total expenses E, category-wise totals, monthly cash flow and the savings rate

$$SR = (I - E) / I \times 100\% \quad (2)$$

These values populate the dashboard and form the context supplied to the recommendation module and the chatbot.

F. Anomaly Detection

Unusual transactions are detected with Isolation Forest [14], an unsupervised method that isolates observations by random partitioning; anomalies have shorter average path lengths $h(x)$ and receive higher scores

$$s(x, n) = 2^{(-E[h(x)] / c(n))} \quad (3)$$

where $c(n)$ is the average path length of an unsuccessful search in a binary search tree of n samples. Input features are the amount, category code, day of week and the amount's z-score within its category. Flagged transactions generate alerts on the dashboard.

G. Expense Forecasting

Monthly expenses are forecast from historical totals. The default model is a linear regression with trend and annual seasonality,

$$\hat{y}(t) = \beta 0 + \beta 1 \cdot t + \beta 2 \cdot \sin(2\pi t/12) + \beta 3 \cdot \cos(2\pi t/12) + \beta 4 \cdot f(t) \quad (4)$$

where t is the month index and $f(t)$ is an optional indicator for festive or other recurring high-spend months. Other models, such as random forests [17], can be substituted through the same interface.

H. Recommendations and Chatbot

The recommendation engine compares actual spending with the forecast and the user's income to propose category-wise budget limits and saving targets, and raises alerts when a category approaches its limit. The chatbot uses an LLM API to answer questions such as where most money is going. To limit exposure of raw personal data, the design passes the model aggregated statistics, forecasts and flagged items rather than the complete statement, and instructs it to answer only from this context.

V. IMPLEMENTATION AND REQUIREMENTS

The system is implemented in Python 3.10+ with Django as the back-end framework. Pandas and NumPy handle data processing, Scikit-learn provides the ML models [15], and Matplotlib generates charts. The front end uses HTML, CSS and JavaScript (or React), and data are stored in MySQL or PostgreSQL. Development uses Visual Studio Code, Google Colab, Git and GitHub. Table II lists the software and hardware requirements.

Data sources for development and testing include sample financial transaction datasets, publicly available datasets where permitted, user-provided CSV statements, historical records supplied by users, and generated synthetic transaction data. Fig. 3 shows an illustrative view of the planned dashboard populated with synthetic data.

TABLE II
Software and Hardware Requirements

Component	Specification
Operating system	Windows 10 or Ubuntu 22.04 LTS
Language / back end	Python 3.10+; Django
Front end	HTML, CSS, JavaScript or React
Libraries	Pandas, NumPy, Scikit-learn, Matplotlib
Database	MySQL or PostgreSQL
AI / LLM	Suitable AI model or API
Tools	Visual Studio Code, Google Colab, Git, GitHub
Processor	Intel Core i5 / AMD Ryzen 5 or higher
Memory / storage	8 GB RAM minimum (16 GB recommended); 256 GB SSD
Network	Stable Internet for cloud AI/API services

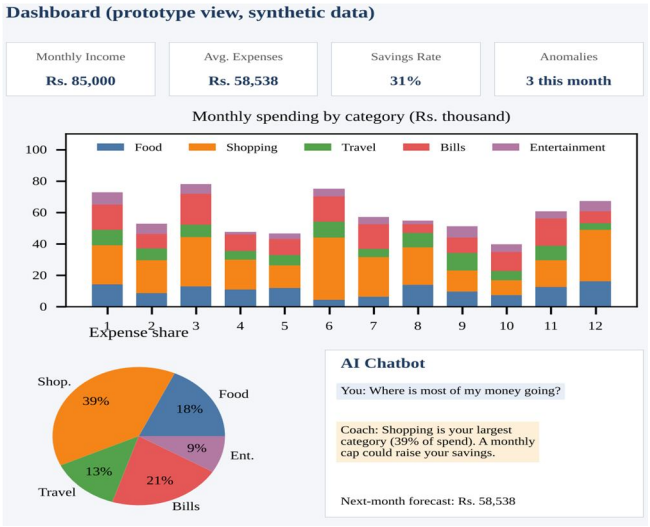


Fig. 3. Illustrative prototype view of the dashboard and chatbot, rendered with synthetic data (not real user data).

VI. PRELIMINARY EXPERIMENTAL RESULTS

A. Experimental Setup

Because real bank statements contain private information, the AI core was evaluated offline on synthetic data generated with a fixed random seed. All experiments used Python and Scikit-learn [15]. The datasets mimic common patterns but are simplified, so the results below indicate feasibility rather than real-world performance.

Categorization dataset. 2,400 transactions (480 per category) were generated from 8–10 merchant description templates per category, with random payment prefixes (e.g., UPI/, POS), reference numbers, 15% upper-casing, 25% truncated merchant names and 6% random label noise. The data were split 80/20 with stratification (1,920 training, 480 test). Features were TF-IDF word unigrams and bigrams; Naive Bayes, linear SVM and logistic regression were compared.

Anomaly dataset. 765 transactions over 365 days of 2025 with category-dependent amounts, into which 40 anomalies were injected by multiplying the amount by a factor between 4 and 9. Isolation Forest (200 trees, contamination set to the anomaly rate) was compared with a baseline flagging transactions whose within-category |z-score| exceeds 3.

Forecasting dataset. A 48-month series with linear trend, annual seasonality, higher spending in two festive months and 5% noise. The first 36 months were used for training and the last 12 for testing. Models were compared by MAE, RMSE and

$$MAPE = (100/n)\ \Sigma\ |y(t) - \hat{y}(t)| / y(t) \qquad (5)$$

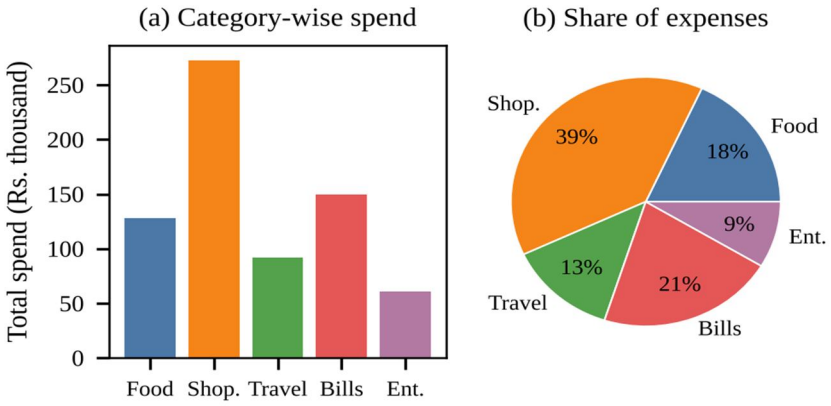


Fig. 4. Category-wise expenditure in the synthetic 2025 transaction set: (a) total spend and (b) share of expenses.

B. Transaction Categorization

Table III shows that the three classifiers perform almost identically on this dataset, at about 92% accuracy; Naive Bayes and logistic regression tie on accuracy (92.29%) while the linear SVM is slightly lower (91.67%). Differences of this size on 480 test samples are not meaningful, so logistic regression is retained for its probability outputs. The confusion matrix (Fig. 5) shows that the largest single confusion is Entertainment predicted as Shopping (7 cases), which is plausible given that merchants such as an online retailer can appear in both categories when the name is truncated.

TABLE III
CATEGORIZATION PERFORMANCE (MACRO-AVERAGED, %)

Model	Acc.	Prec.	Rec.	F1
Naive Bayes	92.29	92.36	92.34	92.30
Linear SVM	91.67	91.72	91.71	91.68
Logistic Regression	92.29	92.38	92.35	92.29

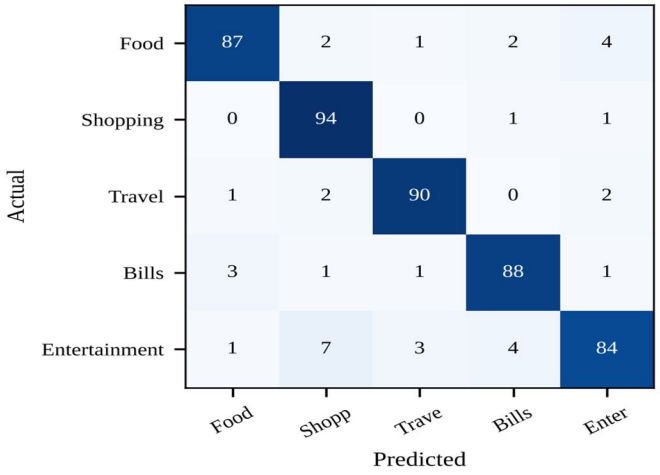


Fig. 5. Confusion matrix of the logistic regression categorizer on the 480-transaction test set.

C. Anomaly Detection

The 40 injected anomalies represent 5.2% of the 765 transactions. Table IV and Fig. 6 show that Isolation Forest detects 37 of the 40 (recall 0.925) with equal precision, whereas the z-score rule is perfectly precise but misses almost half of them (recall 0.525), because large injected amounts in high-variance categories such as Shopping may not exceed three standard deviations. Isolation Forest therefore gives a considerably higher F1-score (0.925 versus 0.689).

TABLE IV
ANOMALY DETECTION PERFORMANCE

Method	Precision	Recall	F1
Isolation Forest	0.925	0.925	0.925
Z-score (|z| > 3)	1.000	0.525	0.689

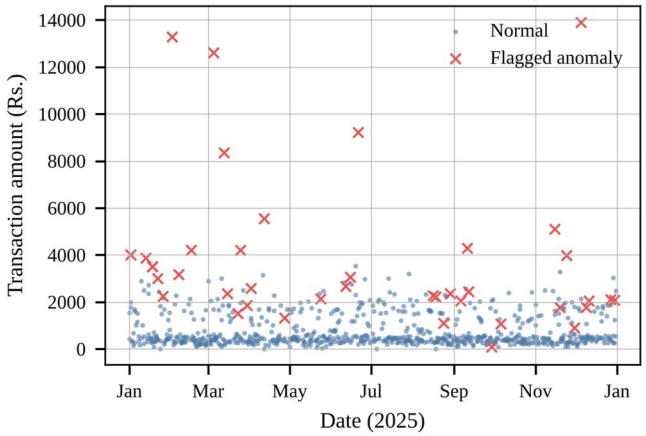


Fig. 6. Transactions flagged by Isolation Forest (red crosses) over the year.

D. Expense Forecasting

Table V and Figs. 7–8 compare the forecasters on the 12 held-out months. The seasonality-aware linear regression gives the lowest error (MAPE 4.19%), compared with 10.40% for plain linear regression, 9.92% for random forest and 12.46% for a three-month moving average. The random forest cannot extrapolate beyond the range of its training targets, so it underestimates the rising trend. Note that the synthetic series was generated with a sinusoidal seasonal component and a festive-month flag that the regression model is given as features, so this experiment favours that model; real data will be less regular.

TABLE V
FORECASTING ERROR ON 12 HELD-OUT MONTHS

Model	MAE (INR)	RMSE (INR)	MAPE (%)
Moving Average (3)	3,380	4,209	12.46
Linear Regression	2,770	3,398	10.40
Linear Reg + Seasonality	1,103	1,306	4.19
Random Forest	2,583	2,843	9.92

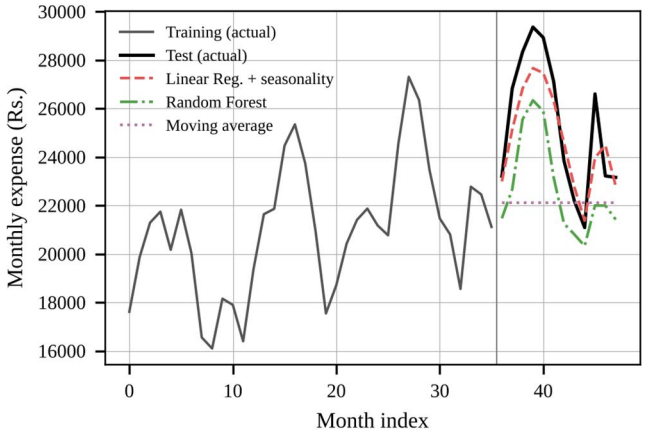


Fig. 7. Actual and forecast monthly expenses on the test period.

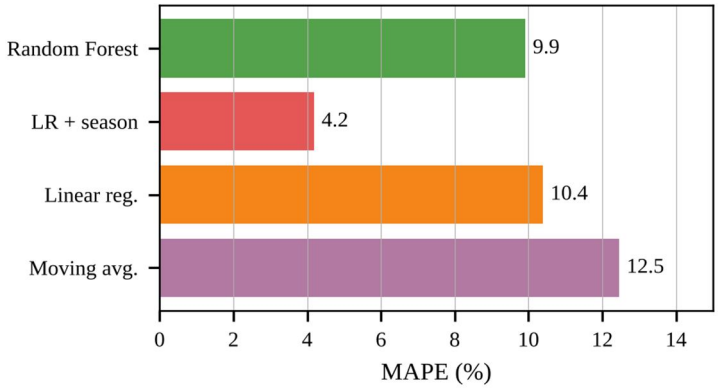


Fig. 8. Forecasting error (MAPE) of the compared models.

VII. DISCUSSION AND LIMITATIONS

The experiments indicate that standard, lightweight ML techniques are sufficient for the categorization, anomaly detection and forecasting stages of the proposed pipeline, and that simple baselines are clearly outperformed for anomaly detection and forecasting. However, several limitations apply. First, all results come from synthetic data with known structure; real statements contain ambiguous merchants, UPI identifiers, mixed languages and irregular income, which will lower accuracy. Second, the dashboard in Fig. 3 is an illustrative rendering, and the full web application, LLM assistant and recommendation engine have not yet been evaluated with users. Third, sending financial summaries to an external LLM API raises privacy and security concerns that require encryption, authentication, consent and data-minimization measures. Fourth, the chatbot's answers must be checked for factual consistency with the computed statistics before being shown.

Future work includes validating the pipeline on anonymized real statements with user consent; integrating bank or UPI APIs for real-time data; adding OCR-based receipt processing [11]; supporting multiple currencies and languages; building a mobile application; improving personalization of recommendations; and conducting a usability study with real users.

VIII. CONCLUSION

This paper presented the design of an AI-powered personal finance coach that unifies CSV-statement ingestion, data cleaning, automatic categorization, spending analysis, anomaly detection, expense forecasting, personalized recommendations, dashboards and a conversational assistant. A preliminary offline evaluation on synthetic data achieved about 92% categorization accuracy, an anomaly-detection F1-score of 0.925 and a forecasting MAPE of 4.19%, outperforming simple baselines for the last two tasks. The modular architecture allows each component to be improved independently. Evaluation on real data and a complete user study are the next steps.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

IX. FUTURE WORK

In the future, this project can be improved in several ways. The system can be tested with real bank statements, with the users' permission, to check how well it works in real life. The web application, budget suggestions and chatbot can be completed and tried by real users to see how much they help with saving money. Instead of uploading CSV files, the system can be connected to bank and UPI services so that transactions are added automatically. Users could also add expenses by scanning receipts and bills. More advanced prediction models can make the forecasts more accurate and the advice more personal to each user's goals and spending habits. Stronger security, such as encryption and secure login, can protect users' financial data. Finally, the system can support more languages and currencies, and a mobile app can make it easier to use every day.

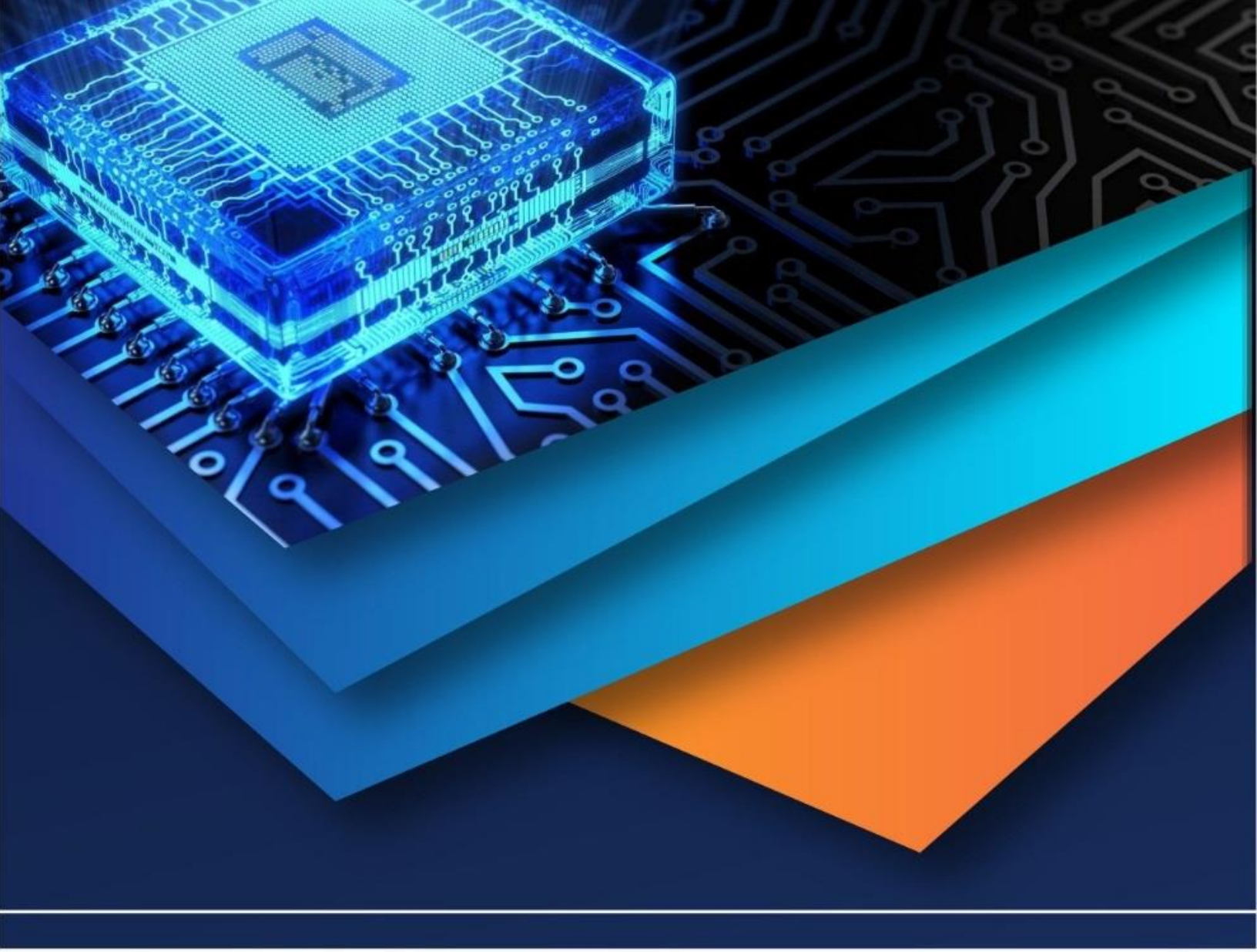
X. ACKNOWLEDGMENT

The authors thanks to their Project Guide Prof. Vidya B. Kale , the project coordinator Prof. Suvarna S. Wakchaure , and the Head of Department of Computer Engineering, Dr. Rahul M. Dhokane ,for their support and guidance during this work.

REFERENCES

[1] B. Chaugule, V. Pawar, O. Chavan, S. Gaikwad, and S. Kachare, “AI-driven personal finance and investment recommendation system,” 2026.
[2] Divya S, M. Inzamam S, Mukeesh R, and Praveen C, “AI-driven personal finance assistant with chatbot,” 2026.
[3] Manasa S, S. K. Shivashankar, P. Prasanna, and H. P. Mohan Kumar, “AI based personal finance management system,” Int. J. Eng. Res. Technol. (IJERT), vol. 14, no. 8, 2025.
[4] M. Sahel, S. Ismail, S. Sohail, and S. Sirasawada, “Streamlining personal finances: An expense tracker website study,” in Proc. IEEE Int. Conf. Signal Processing and Advanced Research in Computing (SPARC), 2024.
[5] L. Lingayat, N. Yadav, P. Rathod, and P. Durutkar, “Design and implementation of real-time expense tracker using machine learning,” in Proc. Int. Conf. Innovative Computing and Communication (ICICC), 2024.
[6] V. Agarwal, R. Ray, and N. Varghese, “An AI-powered personal finance assistant: Enhancing financial literacy and management,” 2024.
[7] S. D. Talasila, “AI-driven personal finance management: Revolutionizing budgeting and financial planning,” Int. J. Eng. Res. Technol. (IRJET), vol. 11, no. 7, 2024.
[8] G. Chauhan, Y. Chaprana, D. Singh, and V. Kumar, “SmartExpense: A CNN-enhanced personal finance tracker with anomaly detection,” Int. J. Res. – GRANTHAALAYAH, vol. 12, no. 1, 2024.
[9] N. Kulshrestha, “Real-time expense tracking and financial forecasting through AI and machine learning,” Int. J. Sci. Res. Eng. Manag., vol. 9, no. 5, pp. 1–9, 2025.
[10] S. Aishwarya and S. Hemalatha, “Smart expense tracking system using machine learning,” in Proc. Int. Conf. Artificial Intelligence for Internet of Things (AI4IoT), 2023, pp. 634–639.
[11] Y. Wang, L. Chen, and S. Xu, “AI-based OCR techniques for financial document processing,” IEEE Trans. Comput. Social Syst., vol. 10, no. 6, pp. 1482–1493, 2023.
[12] A. Narayan, S. Kumar, and D. Singh, “AI-driven personal finance assistants: A review of intelligent budgeting and expense automation,” IEEE Access, vol. 11, pp. 89321–89335, 2023.
[13] S. Subha, J. Pandiyan S, K. E, and N. Ahmed S, “AI-powered platform for personal finance,” 2026.
[14] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation forest,” in Proc. IEEE Int. Conf. Data Mining (ICDM), 2008, pp. 413–422.
[15] F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.
[16] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” Inf. Process. Manga. vol. 24, no. 5, pp. 513–523, 1988.
[17] L. Breiman, “Random forests,” Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)