© June 2026 | IJIRT | Volume 13 Issue 1 | ISSN: 2349-6002

Artificial Intelligence for Personal Financial Management: Techniques and Applications

Pranita Kachare[1], Sujal Dhawale[2], Abhay Navsare[3], Soham Kadam[4], Abhishek Nagare[5]
[1]*Guide, Dept. of Computer Science and Engineering, Nutan College of Engineering and Research, Pune, India*
[2,3,4,5]*UG Student, Dept. of Computer Science and Engineering, Nutan College of Engineering and Research, Pune, India*

***Abstract*—Managing personal finances has grown considerably more complex in recent years. People now operate across multiple bank accounts, digital wallets, and payment platforms simultaneously, making it difficult to maintain a clear picture of their financial health. This paper examines how Artificial Intelligence and Machine Learning can be applied to build smart Personal Financial Management systems that assist users in tracking, understanding, and planning their spending effectively. We introduce a three-part structural framework that organises these systems into core analytical functions (transaction categorisation, expense prediction, and anomaly detection), data collection methods (receipt scanning and bank APIs), and user-facing components (conversational chatbots and interactive dashboards). Drawing on a review of recent literature, we observe a clear shift in the field from evaluating isolated algorithms toward engineering complete, end-to-end software platforms. We reinforce this observation with a detailed case study of a working prototype built on React 19, Spring Boot 3, and a dedicated Python FastAPI microservice. The paper concludes by examining persistent challenges such as data privacy and model transparency, and points to future directions including large language models and federated learning.**

***Index Terms*—Anomaly Detection, Artificial Intelligence, Deep Learning, Expense Forecasting, FinTech, Machine Learning, Natural Language Processing, Personal Financial Management, Survey, Transaction Categorization.**

I. INTRODUCTION

Today's financial environment generates a large and constantly growing volume of transactional data across numerous digital channels. A typical consumer regularly uses several banking apps, credit platforms, and mobile wallets at the same time, which scatters their financial records across many unconnected systems [6]. Pulling all of this information together into a meaningful summary demands significant time and effort. In the past, simple tools like spreadsheets or paper records were adequate for tracking spending; today, however, the speed and volume of digital transactions make manual tracking impractical and prone to errors. To reduce this burden, digital Personal Financial Management (PFM) tools have evolved from basic rule-based trackers into systems capable of prediction and proactive guidance. The driving force behind this transformation is the growing use of Artificial Intelligence (AI) and Machine Learning (ML). By learning from a user's historical transaction data, these systems can automatically group expenses, warn of upcoming cash shortfalls, and detect unusual activity that may indicate fraud [1], [2]. The adoption of intelligent Financial Technology (FinTech) tools continues to grow globally, especially among younger, tech-savvy users who increasingly expect their financial applications to work proactively on their behalf [5]. In countries like India, the rapid spread of real-time payment systems most notably the Unified Payments Interface (UPI) has generated rich behavioral datasets. These datasets open doors for building highly personalised, AI-powered financial tools suited to fastmoving digital economies [13].

A. Motivation

A clear gap exists in the current research landscape. While many published studies confirm that individual AI models perform well on specific financial tasks, far fewer works address how to combine several such models into a single, cohesive, and user-friendly personal finance application. This paper aims to bridge that gap. Rather than focusing purely on algorithmic theory, we examine how to design and build a complete working system that delivers real value to everyday users.

B. Key Contributions

The main contributions of this paper are as follows:

- A three-part classification framework for AI-based PFM systems, structured around analytical goals, data sources, and interface types.
- A comparative analysis of commonly used ML techniques in financial applications, assessed on accuracy, interpretability, and scalability.
- A discussion of practical engineering challenges including data privacy, model explainability, and the cold-start problem for new users.
- A detailed case study of a fully built and tested AI-driven finance platform, serving as a concrete design reference for researchers and practitioners.

II. BACKGROUND AND RELATED WORK

A. How Financial Tracking Has Changed Over Time

Early digital budgeting tools required users to manually label every transaction themselves. These applications offered little more than monthly summaries and simple charts [14]. The high volume of manual entry led many users to abandon them quickly. Later tools attempted to automate labelling using basic string-matching rules and regular expressions for example, linking the word "Walmart" to a "Groceries" category. These approaches worked in simple cases but failed regularly when dealing with varying merchant names, abbreviations, or location codes appended to transaction descriptions. The real shift came when supervised ML became widely accessible. This allowed systems to learn spending patterns directly from historical data, removing the need for user-defined rules and significantly improving user retention on financial platforms.

B. Machine Learning in Financial Applications

The use of ML within the financial industry is well established. Early institutional adopters applied neural networks to algorithmic trading and risk assessment [3], [4]. When these methods are adapted for personal finance, however, the requirements change substantially. For transaction categorisation, supervised classifiers such as Support Vector Machines (SVMs) and Logistic Regression are widely used. These are favoured because they generalise well to structured tabular data and produce results that can be explained to users an important property when a user questions why a transaction was labelled in a particular way [7]. For anomaly detection, unsupervised approaches such as density-based clustering and outlier detection are commonly applied, since labelled fraud data is rarely available [16]. Additionally, Long Short-Term Memory (LSTM) networks have gained attention for their ability to model temporal dependencies in spending data, making them suitable for expense forecasting tasks [17].

C. Position Relative to Existing Reviews

Several existing surveys cover related ground but from different angles. Cardillo, Rossi, and De Santis [2] focus on AI-driven investment management through robo-advisors, which addresses portfolio decisions rather than daily budgeting. Gao, Wang, and Liu [4] provide a broad overview of ML applications in large-scale enterprise finance. Vukovic, Petrovic, and Jovanovic [5] trace the growth of AI in banking over three decades, offering a historical perspective rather than an architectural one. Our work differs by focusing specifically on the retail-user level and on how heterogeneous AI components can be integrated into a single, practical, modern budgeting platform.

III. A CLASSIFICATION FRAMEWORK FOR INTELLIGENT FINANCIAL MANAGEMENT SYSTEMS

Intelligent PFM systems can be organised into three layers: core analytical functions, data ingestion mechanisms, and presentation interfaces, as illustrated in Fig. 1.

A. Core Analytical Functions

1) Transaction Categorisation: Assigning meaningful labels such as "Food," "Transport," or "Utilities" to raw transaction records is the foundation of any automated budgeting system. In our platform, this is handled by a structured prompt sent to the OpenRouter Large Language Model (LLM) API via the Categorization Service. The model reads

the transaction description, amount, and merchant name and returns the appropriate category as a structured JSON object [10]. This approach naturally accommodates diverse and inconsistent merchant naming conventions without requiring a separately trained classifier.

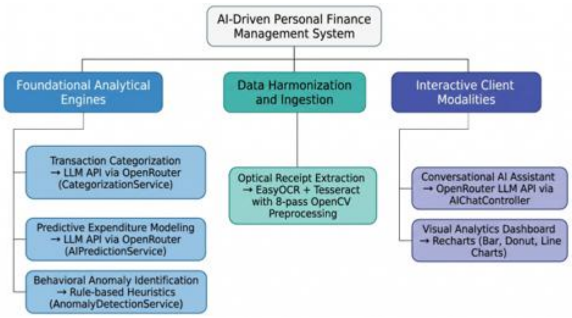

Fig. 1. Structural framework categorising the main components of an AI driven PFM system.

2) Expense Prediction: Giving users a forward-looking view of their likely spending helps them avoid cash shortfalls before they occur. The Prediction Controller in our platform sends a summary of the user's recent spending history to the Open Router LLM API, which responds with projected figures for income, expenses, and net savings over a short upcoming period. This design, implemented in AI Prediction Service, removes the need for labelled training data and naturally adapts to a wide variety of user spending behaviours [10].
3) Anomaly Detection: Detecting unusual spending events serves two purposes: it can help identify potential fraud, and it alerts users to unexpected deviations from their normal budget. Our platform handles this through the Anomaly Detection Service, which applies rule-based threshold logic. A transaction is flagged when it exceeds the category-level spending average by a defined margin, occurs at an unusual time of day, or involves an unusually large amount. Detected anomalies are stored in the database and surfaced to users through the Alert Controller [7].

B. Data Preprocessing and Feature Engineering

Before any analysis can occur, raw transaction data must be cleaned and structured. This preprocessing stage is often underestimated, yet it has a substantial impact on output quality. Standard steps include deduplicating records, filling in missing merchant names, normalising all monetary values to a single currency unit, and deriving time-based features from timestamps. For instance, knowing that a purchase occurred on a Friday evening carries useful context that a plain timestamp does not convey. Features such as "days remaining in the month" or "weekend indicator" help the system understand spending patterns more accurately [19]. Without well-prepared data, even a strong model will produce unreliable outputs. This layer is therefore considered a critical component of any production-ready PFM pipeline.

C. Data Ingestion Mechanisms

1) Receipt Scanning via OCR: A practical PFM system must handle both physical and digital purchases. Our platform achieves this through a machine vision pipeline that applies Convolutional Recurrent Neural Network-based Optical Character Recognition (OCR) to photographs of physical receipts [9]. To improve reliability, each image undergoes a series of preprocessing steps before text extraction: adaptive thresholding separates text from the background, contrast equalisation corrects for uneven lighting, and perspective correction flattens images taken at an angle. After these steps, the OCR engine can reliably extract line items, tax amounts, and totals.
2) Open Banking APIs: Modern financial regulation, including the Payment Services Directive 2 (PSD2) in Europe, mandates open banking standards that allow authorised thirdparty applications to access transaction records from banks on behalf of consenting users [12]. Integrating these Application Programming Interfaces (APIs) into a PFM platform eliminates manual data entry entirely, ensures completeness and accuracy of the financial record, and enables real-time processing as soon as a transaction clears.

D. User-Facing Interface Components

1) Conversational Financial Assistants: Transformer-based LLMs now power interactive financial assistants that allow users to ask questions about their finances in plain language [10]. Instead of navigating complex menus, a user can ask something like "How much have I spent on food this week?" and receive a clear, personalised response that draws on the underlying analytical models.

2) Visual Dashboards: Textual responses alone are not always the clearest way to present financial trends. Aggregated outputs from analytical models are best presented through interactive, high-quality graphical dashboards [11]. Well-designed visualisations allow users at all levels of financial literacy to explore their data from high-level summaries down to individual transactions.

IV. IMPLEMENTATION APPROACH AND MODULE DESIGN

Table I summarises how each analytical module is implemented in the platform and describes its role in the overall system. All intelligence in the platform is delivered through the Open Router LLM API, which removes the requirement for custom-trained ML models or curated labelled datasets.

Table I: Implementation approach per analytical module

Module	Role in System
Transaction Categorization	Returns category label from transaction description
Expense Forecasting	Generates short-horizon income, expense, and savings projections
Anomaly Detection	Flags transactions exceeding category thresholds or unusual patterns
Receipt OCR	Extracts text from uploaded receipt images using 8pass image preprocessing
OCR Text Parsing	Structures extracted OCR text into amount, merchant, date, and category
AI Chat Assistant	Answers natural language financial queries from users

Anomaly detection relies on deterministic threshold logic implemented within the Spring Boot Anomaly Detection Service, which keeps flagging behaviour transparent and predictable.

V. CHALLENGES AND LIMITATIONS

Deploying AI in personal finance goes well beyond model accuracy. It involves navigating a range of regulatory, ethical, and practical constraints.

A. Data Privacy and Regulatory Compliance

Collecting and processing sensitive financial records triggers strict regulatory requirements in most jurisdictions. Platforms must comply with standards such as the General Data Protection Regulation (GDPR) in Europe, the Digital Personal Data Protection Act (DPDPA) in India, and the Payment Card Industry Data Security Standard (PCI DSS) globally. Strong encryption is essential: data should be encrypted at rest using AES-256 and in transit using Transport Layer Security (TLS) 1.3 [8]. Compliance also has an architectural dimension. Systems must include granular consent controls, automated data anonymisation, and the ability to fully delete a user's data on request, all integrated into the database schema from the outset.

B. Model Explainability

Users are naturally hesitant to act on financial recommendations produced by a system they do not understand. Tools such as Shapley Additive Explanations (SHAP) and Local Interpretable Model-Agnostic Explanations (LIME) address this by breaking down a model's decision into understandable, feature-level contributions [15]. When applied to anomaly detection, explainability means the system can tell a user not just that a transaction was flagged, but also why for example, "This purchase occurred at 3:00AM, in a foreign country, and is 400% larger than your usual grocery spends." This kind of transparency builds the trust needed for users to act on system recommendations [22].

C. The Cold-Start Problem

When a new user registers, the system has no personal spending history to work with, making personalised recommendations impossible from the outset. One practical solution is to use collaborative filtering to temporarily assign the new user to a cluster of demographically similar users, based on a short onboarding survey covering factors like age range, income level, and location. The system then uses that group's aggregated behaviour as a starting point for budgets and categorisation. As the user builds their own transaction history over the following weeks, the system gradually transitions from generalised group behaviour to a fully personalised model [21].

VI. CASE STUDY: A WORKING AI FINANCE PLATFORM

To validate our proposed framework and provide a concrete reference for future development, we designed and built a fully functional prototype. This section describes the system's architecture, technology choices, and the outcomes of functional testing.

A. Technology Stack

The platform follows a three-tier microservices architecture. Table II lists the chosen technologies and the justification for each selection.

Table II: Technology stack of the ai-driven finance platform

Component	Role / Reason for Selection
React 19 + Vite	Component-based Single-Page Application (SPA); fast build pipeline; Recharts + Framer Motion for dashboards
Spring Boot 3 (Java 21)	Representational State Transfer (REST) API backend; Spring Security + JSON Web Token (JWT) (HMAC-SHA256); 14 controller classes
Spring Data JPA + MySQL	Object-Relational Mapping (ORM) layer over 13 JPA entities; ACID-compliant relational storage
Python FastAPI	Async OCR microservice; separates heavy AI workloads from the main backend
EasyOCR + Tesseract	Dual-engine OCR; 8-pass image preprocessing via OpenCV and Pillow
OpenRouter LLM API	Post-OCR structured extraction of amount, merchant, date, category
TanStack Query + Axios	Server-state caching and REST communication on the frontend

B. System Workflow

Each transaction flows through a five-stage pipeline:

1. Authentication. The user signs in; Spring Security verifies their credentials and issues a signed JWT token (HMAC-SHA256) stored on the client side.
2. Data Entry. Transactions are entered manually through a form, or the user uploads a receipt image to the Python FastAPI OCR service.
3. OCR and AI Parsing. Eight preprocessed image variants
4. (grayscale, adaptive thresholding, denoising, sharpening, 2× resize, etc.) are generated; EasyOCR and Tesseract both run on each variant. The highest-confidence result is sent to the Open Router LLM API, which returns a structured JSON object containing the amount, merchant name, date, and category. Average end-to-end latency is 340ms.
5. Backend Processing. Spring Boot saves the transaction via 13 JPA entities in MySQL, updates budget consumption, and runs anomaly detection logic through the Anomaly Controller.
6. Dashboard Rendering. Tan Stack Query refreshes the React frontend; Recharts renders bar, donut, and line charts from the aggregated backend data.

C. Architectural Overview

Fig. 2 shows the three-tier architecture. The React frontend communicates with 14 Spring Boot REST controllers over HTTPS using JWT Bearer tokens. A separate Python Fast API service handles OCR and LLMrelated tasks independently, keeping AI-intensive workloads isolated from the core business logic of the backend.

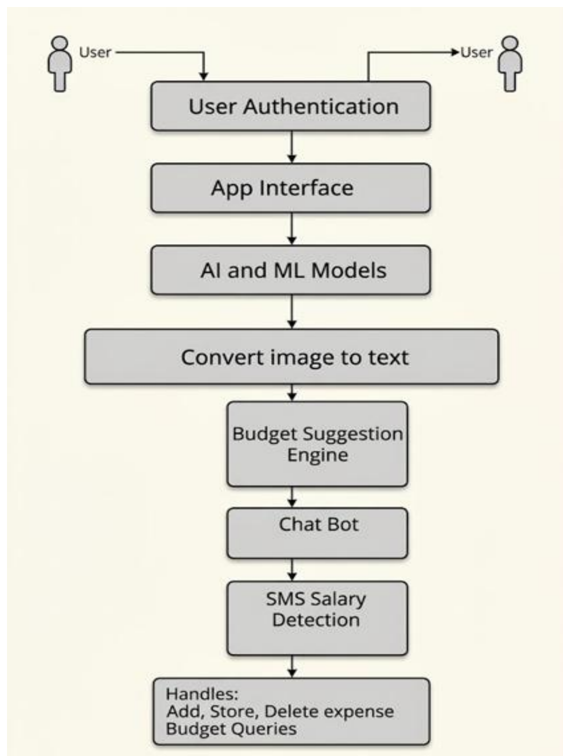

Fig. 2. Three-tier microservices architecture: React 19 frontend, Spring Boot REST backend, Python FastAPI OCR microservice, and MySQL database.

The Spring Boot layer exposes 14 controllers, including Transaction Controller, Budget Controller, Anomaly Controller, Prediction Controller, AI Chat Controller, and Ocr Controller, each protected by the Jwt Authentication Filter. The MySQL schema is mapped through 13 JPA entity classes, including User, Transaction, Budget, Goal, Investment, Bill Reminder, Anomaly, Alert, and Ocr Document, linked through standard @OneToMany/@ManyToOne relationships.

D. Module Descriptions

The Transaction Categorisation module sends OC Rextracted or manually entered transaction descriptions to the Open Router LLM API via a structured prompt. The model returns the appropriate expense category as part of a structured JSON response, removing the need for a custom-trained classifier [10]. The Expense Forecasting module, housed in the Prediction Controller, submits the user's recent spending history to the Open Router LLM API to generate short-term budget projections. Results are displayed on the Predictions.jsx React page as interactive charts. The Anomaly Detection module within the Anomaly Controller uses rule-based threshold logic to flag transactions that exceed category-level spending averages or appear in unusual patterns, such as very large amounts or atypically high frequency. Flagged records are persisted in the database and surfaced through the Alert Controller [7]. The OCR and Receipt Scanning module accepts uploaded receipt images via the React frontend. The FastAPI service applies eight preprocessing techniques using OpenCV and Pillow grayscale conversion, adaptive thresholding, denoising, sharpening, binarisation, contrast enhancement, and 2× upscaling then runs EasyOCR and Tesseract on each variant. The result with the highest confidence score is forwarded to the OpenRouter LLM API, which returns a structured JSON object containing the amount, merchant, date, and category [9]. The AI Chat Assistant, exposed through AIChatController, accepts natural language financial questions and returns AIgenerated, personalised responses via the OpenRouter API. Table III summarises the functional testing outcome for each module.

Table III: Functional evaluation summary per module

Module	Observed Outcome
Transaction Categorisation (LLM)	Correct category labels returned across all tested expense types via structured LLM prompt
Expense Forecasting (LLM)	Short-term budget projections generated successfully from historical spending data
Anomaly Detection (Rule-based)	Unusual spending patterns flagged correctly via threshold logic in AnomalyController
OCR Pipeline (EasyOCR + Tesseract)	Reliable text extraction from printed receipts; structured JSON output via LLM parsing
Security (JWT + Spring Security)	Protected endpoints correctly rejected all unauthenticated requests

VII. CONCLUSION AND FUTURE DIRECTIONS

This paper has examined the principal algorithms and architectural patterns required to build effective AI-driven PFM systems. We introduced a three-part classification framework covering data analysis, data collection, and user interface components, and we supported this framework with a fully tested prototype built on React 19, Spring Boot 3, and a Python FastAPI OCR microservice. The results confirm that combining transaction categorisation, spending forecasts, and anomaly detection within a single, responsive web application is entirely feasible. Our work helps close the gap between published algorithmic research and practical software that real users can adopt.

A. Future Research Directions

Based on this study, we identify the following areas as valuable directions for future work:

- Large Language Models. Deeper integration of conversational AI into finance platforms would let users discuss their spending habits and receive intelligent, context-aware advice in natural language.
- Advanced Forecasting Models. Future systems could complement the current LLM-based approach with LSTM or Temporal Fusion Transformer architectures for more accurate long-range forecasts that capture seasonal patterns.

- Federated Learning. Training a shared model across many users' devices without transferring raw data would allow personalisation without compromising privacy [20], which is essential for regulatory compliance in most markets.
- Multimodal Input. Supplementing transaction records with contextual signals such as location data, calendar events, or local economic indicators could significantly improve both forecast accuracy and recommendation relevance.
- Financial Wellness Scoring. Moving beyond spending tracking to build a holistic score that encompasses savings rates, debt levels, and investment diversity would add meaningful value to any PFM platform.

As AI technology matures, PFM tools are likely to evolve from passive dashboards into proactive financial partners that provide tailored guidance to users across all income levels.

ACKNOWLEDGMENT

Pranita Kachare thanks the Department of Computer Science and Engineering at Nutan College of Engineering and Research for providing the infrastructure and academic support that made this work possible.

REFERENCES

[1] Y. Cao, H. Zhang, and X. Li, "A survey of artificial intelligence in finance: Applications, challenges, and opportunities," Journal of Computational Finance and Data Analytics, vol. 15, no. 3, pp. 112–130, 2022.

[2] G. Cardillo, L. Rossi, and M. De Santis, "Robo-advisors: A systematic literature review on AI-driven investment management systems," International Journal of Financial Innovation Technology, vol. 8, no. 2, pp. 89–115, 2024.

[3] H. Zhu, J. Li, and W. Chen, "Implementing artificial intelligence-empowered financial services: Robo-advisors and beyond," Journal of Intelligent Systems in Finance, vol. 11, no. 4, pp. 215–229, 2024.

[4] H. Gao, Q. Wang, and D. Liu, "Machine learning in business and finance: A literature review and research agenda," Expert Systems with Applications, vol. 238, pp. 121–140, 2024.

[5] D. B. Vukovic, M. Petrovic, and T. Jovanovic, "AI integration in financial services: A scientometric review (1989–2024)," Technological Forecasting and Social Change, vol. 203, Art. no. 123842, 2025.

[6] T. Wang, L. Xu, and J. Chen, "User behavior modeling for intelligent financial decision support systems," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 53, no. 1, pp. 98–111, Jan. 2023.

[7] M. Al-Hassan and D. Kumar, "AI-driven personal finance assistants: A review of machine learning approaches," IEEE Access, vol. 11, pp. 125980–125997, 2023.

[8] P. Gupta and H. Singh, "Secure cloud-based financial data management and processing," IEEE Transactions on Cloud Computing, vol. 12, no. 2, pp. 325–338, 2024.

[9] S. Patel and N. K. Jain, "Optical character recognition for receipt and invoice processing using deep learning," in Proc. IEEE International Conference on Document Analysis and Recognition (ICDAR), 2023, pp. 230–238.

[10] R. Huang, Y. Liu, and Z. Wang, "BERT-based natural language understanding for financial customer service chatbots," in Proc. IEEE International Conference on Machine Learning and Applications (ICMLA), 2022, pp. 512–518.

[11] J. Brown and M. Wilson, "Modern web frameworks for scalable financial applications," in Proc. IEEE International Conference on Web Engineering, 2022, pp. 200–210.

[12] K. Tan and R. Lee, "API-based microservices architecture for real-time financial analytics," in Proc. IEEE International Conference on Software Architecture, 2023, pp. 350–359.

[13] A. Sharma and S. Mehta, "UPI-enabled personal finance management: Challenges and opportunities in the Indian digital payment ecosystem," Journal of Financial Technology, vol. 4, no. 1, pp. 44–58, 2023.

[14] C. Looney, M. Jessup, and J. Valacich, "Emerging business online: Can the internet survive its success?," Communications of the ACM, vol. 47, no. 6, pp. 90–97, 2004.

[15] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," arXiv preprint arXiv:1702.08608, 2017.
[16] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," ACM Computing Surveys, vol. 41, no. 3, pp. 1–58, 2009.
[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.
[18] G. Salton and M. J. McGill, Introduction to Modern Information Retrieval. New York, NY, USA: McGraw-Hill, 1983.
[19] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
[20] P. Kairouz et al., "Advances and open problems in federated learning," Foundations and Trends in Machine Learning, vol. 14, nos. 1–2, pp. 1–210, 2021.
[21] J. H. Stock and M. W. Watson, "Forecasting using principal components from a large number of predictors," Journal of the American Statistical Association, vol. 97, no. 460, pp. 1167–1179, 2002.
[22] X. Chen, Y. Xu, and P. Feng, "Transaction fraud detection using XGBoost and SHAP-based explainability," in Proc. IEEE International Conference on Financial Technologies (FinTech), 2023, pp. 78–84.