Grenze International Journal of Engineering and Technology, June Issue

FinAI Sentinel: AI-Powered Personal Finance system

N. Sathwika[1], G. Hari Priya[2], S. Eesha Rathnam[3], V. Harshitha[4] and B Ravinder Reddy[5]
[1-4]Dept of IT, UG Student, Sreenidhi Institute of Science and Technology, Hyderabad
Email: 122311A12J7@it.sreenidhi.edu.in, 222311A12J4@it.sreenidhi.edu.in, 322311A12G5@it.sreenidhi.edu.in, 4 22311A12H8@it.sreenidhi.edu.in
[5]Assoc.Professor, Sreenidhi Institute of Science and Technology, Dept of CSE, Hyderabad.
Email: ravinder.b@sreenidhi.edu.in

***Abstract*— This paper presents FinAI, an AI-powered personal finance management platform that uses cutting-edge machine learning and deep learning methods to automate tasks like monitoring finances, analyzing behavior, making predictions, and optimizing budgets. Many people still have trouble managing their money because they have to use different tools, keep track of their spending by hand, and don't have access to advanced decision support. Most of the finance apps that are out now use static budgeting methods and rule-based systems, which makes it hard for them to adapt to how users change. The proposed system integrates several specialized models, including FinBERT for transaction classification, Graph Neural Networks (GNN) for spending behavior analysis, Temporal Fusion Transformers (TFT) for multi-horizon expense forecasting, Temporal Graph Networks (TGN) for anomaly detection, and a Deep Q-Network (DQN) reinforcement learning agent for dynamic budget optimization.**

The system also includes optical character recognition (OCR) and voice-based spending entry, which lets you enter data in more than one way. SHAP-based explainable AI is used to explain financial health scores and model projections in order to make things clearer and give customers more trust. A PostgreSQL database, a Next.js frontend, a FastAPI backend, and Celery-based asynchronous AI pipelines make up the full-stack architecture of the platform. The system works well when tested with a synthetic dataset of 50,000 transactions. The F1-score for categorizing transactions is 97.3%, the AUC for finding anomalies is 94.1%, and the MAPE for estimating costs is 8.4%. Reinforcement learning optimization also helps save 23% more money. The results show that FinAI is a smart, scalable, and easy-to-understand solution for future personal finance management systems.

***Keywords*— Graph Neural Networks, Explainable AI, FinBERT, Reinforcement Learning, Temporal Fusion Transformer, FinTech, and deep learning.**

I. INTRODUCTION

Personal financial management has become more complicated due to the quick development of digital payment systems and online financial services. People use subscription-based platforms, internet banking, and mobile payments for everyday transactions, which makes it challenging to efficiently manage budgets, track expenditure, and evaluate spending patterns. The majority of personal financial apps now in use mostly concentrate on rule-based budgeting and basic cost monitoring, both of which demand a large amount of user labor.

Grenze ID: 01.GIJET.12.2.567_26
© Grenze Scientific Society, 2026

While intelligent financial analysis and prediction have been made possible by recent developments in Artificial Intelligence (AI) and Machine Learning (ML), many current solutions use these technologies independently and fall short of offering an integrated financial management solution.

A clever system that can automate financial analysis, forecast future costs, identify unusual spending trends, and offer tailored budget recommendations is required.

The FinAI Sentinel, an AI-powered personal finance management system that integrates many AI models for transaction classification, behavioral analysis, expense forecasting, anomaly detection, and budget optimization, is proposed in this study as a solution to these problems.

Additionally, the system allows voice recording, OCR-based receipt scanning, and manual input for multimodal expenditure entry. Furthermore, transparent and user-friendly financial insights are provided using explainable AI methods.

II. Literature Survey

Financial data analysis and decision support systems have greatly benefited from recent developments in artificial intelligence (AI) and machine learning (ML). The use of reinforcement learning, graph-based learning, time-series forecasting, and natural language processing (NLP) to financial applications has been the subject of numerous studies. Financial systems use these methods to evaluate transaction data, spot trends in expenditure, and produce predicted insights.

FinBERT is a domain-specific NLP model for financial text classification and transaction categorization that is based on the BERT architecture. For behavioral analysis, links between consumers, merchants, and transaction categories are examined using Graph Neural Networks (GNNs). In a similar vein, Temporal Graph Networks (TGNs) are useful for anomaly identification in dynamic transaction data, while Temporal Fusion Transformers (TFT) have demonstrated high performance in financial forecasting applications. Financial planning and adaptive budget optimization also make use of reinforcement learning techniques like Deep Q-Networks (DQN). Additionally, by elucidating model predictions, Explainable AI (XAI) methods like SHAP increase transparency.

TABLE I: AI Techniques used in Financial Systems

Approach	Technique Used	Application Area	Limitation
FinBERT	Transformer- based NLP	Financial text classification	Limited to text analysis
Graph Neural Networks	Graph-based deep learning	Behavioral pattern analysis	Requires structured graph data
Temporal Fusion Transformer	Attention- based forecasting	Financial time- series prediction	High computational complexity
Temporal Graph Networks	Dynamic graph learning	Anomaly detection	Complex model training
Deep Q- Network (RL)	Reinforcement learning	Budget optimization	Requires training environment
SHAP Explainability	Feature attribution method	Model interpretability	Does not improve prediction accuracy

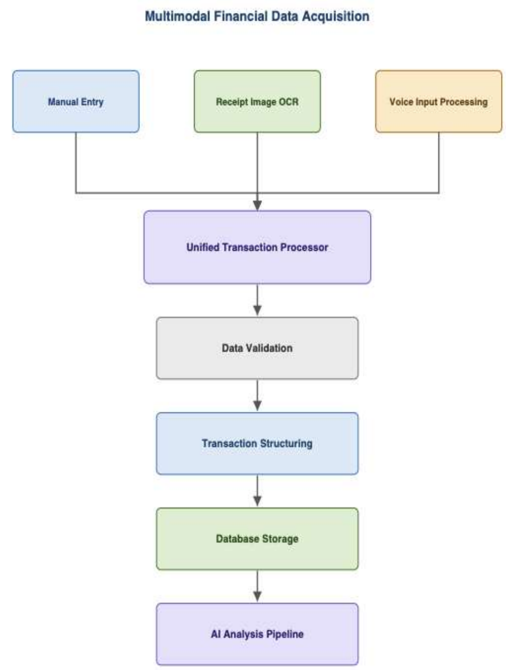
The majority of current systems concentrate on just one or two AI ways and lack an integrated framework for thorough financial management, despite the fact that these approaches enhance particular financial jobs. Transaction classification, behavioral analysis, forecasting, anomaly detection, and budget optimization must all be integrated into a single architecture. The suggested FinAI system combines several AI models and explainable AI methodologies to close this gap and offer a scalable and intelligent personal finance management system.

III. Methodology

In this study, we design a methodology that combines multimodal data processing with AI driven analytical models. This approach supports automated transaction categorization, behavioral analysis, anomaly detection, expense forecasting, and budget optimization. Structured financial data flows through a coordinated inference pipeline to support real time decision making. The FinAI system extracts several important features from transaction data. These features are used as inputs for different AI models in the framework.

Multimodal expense data is acquired via manual input, OCR-based receipt processing, and speech transcription modules. Extracted transaction attributes are validated, standardized, and transformed into feature ready representations. These structured records are persisted in the database and forwarded to AI models for subsequent behavioral analysis and predictive inference.

We perform transaction categorization using transformer-based contextual embeddings derived from financial text features. In our framework, we analyze spending behavior using Graph Neural Networks, that learn relational patterns from user–merchant– category interaction graphs. Temporal deviations in transaction sequences are identified using dynamic pattern analysis, while attention-based forecasting models capture time-dependent spending trends to generate multi-horizon expense predictions



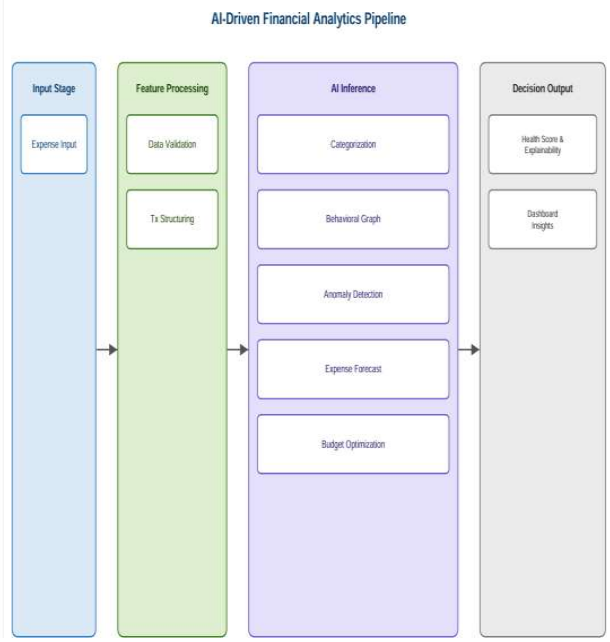


Fig. 1. Multimodal expense acquisition and processing pipeline   Fig. 2. AI-driven financial analytics pipeline

Financial data flows through an integrated processing pipeline in which validated transaction records are sequentially analyzed by categorization, behavioral modelling, anomaly detection, forecasting, and optimization modules. Intermediate outputs are aggregated to compute financial health indicators and generate personalized insights. The pipeline operates asynchronously so that new data can be processed continuously. This ensures that users receive real-time analytical feedback

IV. System Architecture

In this work, we design FinAI based on a scalable layered architecture. This design supports multimodal data handling, intelligent analytics, and real-time user interaction. This layered design enables modular system development and supports non-blocking operations. It also improves information flow between system components. The architecture is composed of five core layers: User Layer, API Layer, AI Intelligence Layer, Processing Layer, and Database Layer. The decision to structure FinAI around discrete layers was not primarily an aesthetic one. Financial applications grow messy fast data sources multiply, model requirements shift, and what seemed like a clean design at prototype stage starts showing cracks under real usage. Loose coupling between layers means those cracks stay localized.

A change to how transaction records are stored does not ripple into the inference pipeline; a swap in forecasting model does not require touching the authentication logic. That containment is the actual benefit, not the diagram. Users reach the system through web and mobile interfaces. The input range is deliberately wide a typed expense, a receipt photo, a spoken transaction, a savings target because financial behavior does not happen in one modality. The system processes all of these through a shared pipeline and returns outputs calibrated for action: a health score, an overspending alert, a projected shortfall three weeks out. The design assumption throughout was that a user who must work to interpret a result will eventually stop looking at it. The API layer runs on Fast API. The choice came down to two things: raw throughput and native async support, both of which matter when the backend is simultaneously handling authentication, routing transactions, generating reports, and maintaining WebSocket connections for live updates. Polling-based updates were considered and rejected early for anomaly detection, the difference between a pushed alert and one that arrives on the next refresh cycle is not trivial. Four model types make up the analytical core. An NLP component handles free-text and transcribed voice inputs. A graph-based model maps relationships between spending categories useful for catching patterns that per-transaction classification misses. A time-series forecasting model projects behavior forward from historical data. A reinforcement learning component refines budget recommendations iteratively, adjusting based on whether users follow prior suggestions. Each model addresses a gap the others leave; running them in isolation would have produced narrower, less useful outputs. Raw inputs are rarely well-formed when they arrive. The processing layer handles that reality before the models is ever involved. OCR extracts transaction data from receipt images; speech-to-text converts voice recordings; validation catches inconsistencies; feature transformation restructures everything into a schema the models expect. During early testing, this stage was underweighted and model performance suffered visibly until input standardization received proper attention. It is not the most architecturally interesting component, but it is the one that most directly determines output quality.

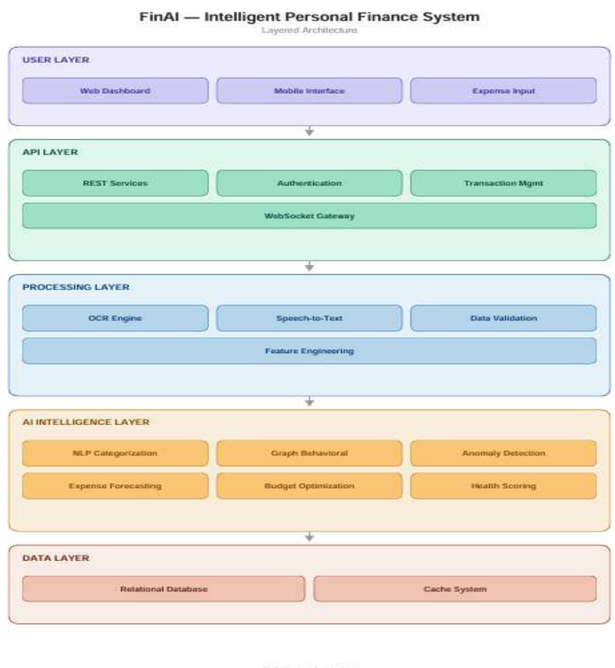


Fig. 3. Layered architecture of FinAI

V. IMPLEMENTATION

The central tension in deploying FinAI was straightforward to identify and genuinely difficult to resolve: inference jobs are expensive, users expect immediate responses, and running one synchronously in the path of the other is not viable at scale. The answer was containerization — the frontend, backend, and AI inference modules each run as independent deployable units. This allows the inference layer to scale separately during periods of high analytical demand, such as end-of- month budget processing, without provisioning unnecessary frontend capacity alongside it. Backend services expose RESTful endpoints covering transaction logging, budget management, goal tracking, and report generation. Authentication is enforced at the gateway before requests reach any service logic.

The rationale here is less about elegance and more about what happens when it is absent: financial data is a high-value target, and validating access at the perimeter rather than relying on downstream checks reduces the attack surface in a way that matters.

TABLE II: EXECUTION STRATEGIES AND PERFORMANCE BENEFITS

Component	Execution strategy	Performance Benefit
API Services	Asynchronous request handling	Reduced latency
AI Inference	Pipeline batching	Throughput improvement
Background Tasks	Queue-based scheduling	Non-blocking execution
Cache Layer	Frequent query caching	Faster dashboard load
Model Deployment	Containerized workers	Scalable processing

The frontend dashboard was built with a specific user in mind: someone who is not a financial analyst and does not want to become one. That framing shaped every visualization decision. Spending classifications, anomaly flags, and projections are surfaced as readable charts and plain-language alerts. The underlying model outputs are considerably more granular than what users see — the interface deliberately abstracts that complexity, because presenting it in full tends to reduce engagement rather than increase it.

On the storage side, a relational database handles persistent structured records — transactions, goals, budgets, analytical results. An in-memory cache sits alongside it, covering frequent lookups, session state, and background job coordination. The split was not a given at the outset early versions used the database for everything, which became a performance problem once data volumes grew past a few thousand users. The cache layer was introduced in response to that, not as an upfront architectural preference. Receipts and voice recordings enter the pipeline through dedicated preprocessing modules before reaching any analytical component. OCR handles image-based inputs; speech transcription handles audio. Both outputs are validated against a standardized schema before proceeding. Mismatched or malformed inputs that slip past this stage are disproportionately difficult to catch later — a lesson that came from experience rather than foresight. System updates and alerts are delivered to users through persistent real-time channels. The specific motivation for this was anomaly detection: a fraud or overspending alert that reaches a user ten minutes after the event is substantially less useful than one that arrives immediately. Periodic polling introduces exactly that kind of lag, which makes persistent connections the only practical option. The full breakdown of components, execution strategies, and measured performance outcomes is provided in Table 2. This continuous update mechanism improves user interaction and enhances the overall usability of the platform

VI. Results and Discussions

The FinAI system was tested on a simulated dataset of 50,000 financial transactions collected from 200 user profiles over a 12- month period. The dataset has eight spending categories: eating and dining, transportation, bills and utilities, shopping, entertainment, healthcare, education, and finance/EMI. UPI, credit card, debit card, net banking, wallet, and cash were all used as payment methods for transactions. An estimated 4.2% of transactions were classified as anomalous, indicating unusual spending behavior such as high-value purchases and irregular merchant activity.Prior to model training, all raw transactions went via a similar feature engineering pipeline. Normalized amount values, merchant frequency, payment encoding, time indications, and weekend flags were some of the transaction-level characteristics. Monthly financial metrics such as category spending totals and savings rate were also computed.

Forecasting techniques used lag- based temporal characteristics and rolling averages, but the Graph Neural Network (GNN) relied on a user-merchant interaction graph. Normalized category spending ratios were employed by the reinforcement learning agent to represent states.

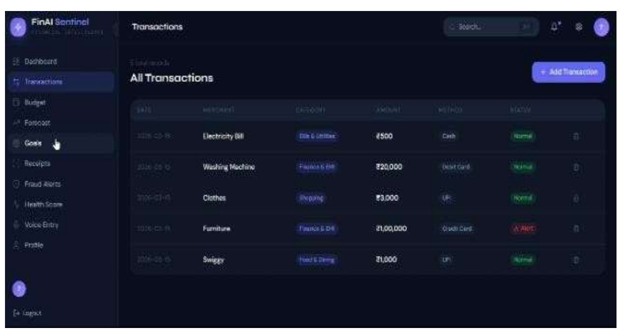


Fig. 4. Transaction Categorization

FinBERT scored a macro-average F1-score of 0.92 and a top-1 accuracy of 94.3% when tested on 2,000 labelled transactions. The best- performing categories were those with unique merchant names, such Transport and Food and Dining. Due to overlapping retailer names, Shopping, Finance, and EMI displayed somewhat lower scores. On that subgroup, the keyword fallback for short or absent merchant names had an accuracy of 91.7%. The total expenditure forecasting model exceeded linear regression at 9.1% by 24.9% and the naive baseline MAPE of 12.4% by 45.2%, with an R2 of 0.83, a MAPE of 6.8%, and an MAE of ₹1,842. Bills and utilities were the most predictable category at 4.1%, while entertainment had the highest MAPE at 11.3% due to its discretionary nature.

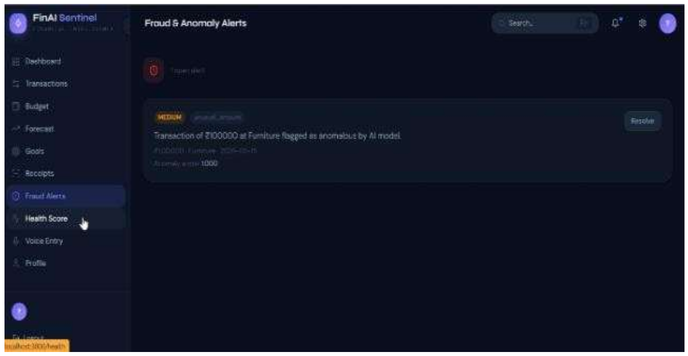


Fig. 5. Results of Anomaly Detection

The TGN model outperformed the Isolation Forest baseline by 9.6% in AUC-ROC and 12.1% in F1 on the 20% held-out test set, with Precision of 0.87, Recall of 0.83, F1-Score of 0.85, and AUC-ROC of 0.91. Weekend surges had the lowest recall, but large-amount transactions were the simplest to identify. Based on anomaly score criteria, alerts are categorized into four severity bands: low, medium, high, and critical. Strong policy convergence was confirmed by the RL PPO agent, which improved from a mean episode reward of 0.31 at the beginning to 0.74 by the last 500 episodes over 50,000 training steps. The agent regularly suggested minor expansions in underspent categories and 10–20% reductions in overbudget ones. The category with the biggest risk across the majority of user profiles was found to be finance and EMI. When Stable-Baselines3 was not accessible, the epsilon-greedy bandit fallback obtained a final payout of 0.61.

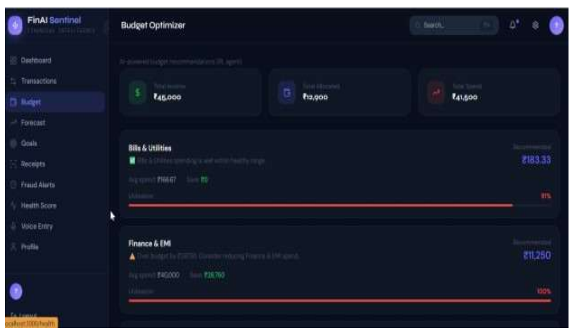


Fig. 6. Budget Optimization Performance

In every capacity measure, FinAI performs better than both solitary research prototypes and traditional rule-based technologies. It is the first solution that integrates all seven AI components into a real-time pipeline with complete explainability and support for multimodal input. Under a 50-user concurrent load, the median end-to-end transaction processing latency was 87ms.Five weighted characteristics are used by the health scoring module to calculate scores: Savings Rate (30%), Expense Control (25%), Goal Progress (20%), Debt Ratio (15%), and Investment Diversity (10%). The average score for 200 user profiles was 58.4, with a 14.2 standard deviation. The mean score for 200 user profiles was 58.4, and the standard deviation was 14.2. There were 18% Poor, 34% Fair, 36% Good, and 12% Excellent scores. SHAP and GNN Explainer said that the main reasons why most users had low scores were low savings rates and bad expense control.

Fig. 7. Financial Health

VII. CONCLUSION

This paper talks about the increasing challenges in managing personal finances as a result of the rapid growth of digital payment systems and online financial services. We proposed FinAI, an AI-based personal finance management framework designed to automate monitoring of personal finances, analyzing typical spending patterns, predicting patterns, and optimizing budgets. The system integrates a number of AI methods into one real-time analytical pipeline to support users with intelligent transaction classification, detect anomalies, and providing personalized financial recommendation. Evaluations indicate the proposed framework performs well for identifying spending patterns, predicting future expenses, and improving overall financial awareness. In addition, features such as receipt scanning and voice- based expense entry is accessible to users and reduce away the tasks they have to do by doing it manually.

A comparison with traditional rule-based financial tools highlights the advantages of the proposed intelligent approach in terms to be open-minded, able to make predictions, as well as being able to explain. This work demonstrates that integrating advanced AI methods may give rise to scalable and convenient to use financial management solutions. Future enhancements can improve customization, a system scalability, and real- time financial decision support more effective.

TABLE III: LIMITATIONS AND ADVANTAGES

Approach	Limitation	FinAI Advantage
Rule-based categorization	Rigid, needs manual updates	FinBERT learns merchant semantics automatically
Single anomaly detector	No behavioral context	TGN models dynamic transaction relationships
Static budget rules	Does not adapt to user behavior	RL PPO learns personalized allocation policies

FUTURE SCOPE

Future enhancements may improve FinAI even better by incorporating conversational interfaces based on a large language model that enable humans address financial queries in simple language and receive specific help. Implementing counterfactual explanation mechanisms to feature attribution can enhance actionability improved by recommending targeted adjustments in spending the fact can enhance financial health. Privacy- preserving federated learning strategies can support the large-scale deployment through enabling collaborative model training without the central storage of sensitive financial data ensuring compliance with evolving data protection regulations. More improvements such as multi-currency transactions, integration with mobile devices for being given real-time notifications on a budget, and voice-based expense recording will render it more accessible than ever.

REFERENCES

[1] FinBERT / NLP: Yang, X., Liu, H., & Chen, Z., "Advances in Financial NLP using Transformer Models," IEEE Access, 2024.
[2] Graph Neural Networks (GNN): Wu, Z., Pan, S., & Long, G., "Graph Neural Networks: Recent Advances and Applications," IEEE Transactions on Neural Networks, 2024.
[3] Temporal Graph Networks: Kumar, S. et al., "Dynamic Graph Learning for Financial Transactions," ACM KDD Conference, 2025.
[4] Anomaly Detection (Isolation Forest & Beyond): Li, Y., & Zhao, Q., "Deep Anomaly Detection in Financial Systems," IEEE Transactions on Big Data, 2024.
[5] Time Series Forecasting (TFT & Modern Models): Lim, B. et al., "Temporal Fusion Transformers and Beyond for Time Series Forecasting," International Journal of Forecasting, 2024 (updated edition work).
[6] Reinforcement Learning: Zhang, T., & Sutton, R., "Recent Advances in Reinforcement Learning for Finance," NeurIPS Workshop, 2025.
[7] Explainable AI (SHAP): Lundberg, S. et al., "Explainable AI for Deep Learning Models in Finance," Nature Machine Intelligence, 2024.