

Privacy-First Local Wealth Intelligence Terminal

Prof. Nitin Mahajan¹, Bharat Sanjay Salve², Nirav Sandip Borde³,
Prajwal Sanjay Desale⁴, Atulkumar Vinodkumar Sah⁵, Mr. Vinod Bhosale⁶
School of CSE¹⁻⁵

Industry Mentor: Senior Solution Architect, Nokia Networks and Solutions⁶
Department of Computer Science & Engineering
Sandip University, Nashik, Maharashtra, India

Abstract: Artificial Intelligence (AI) has revolutionized per-sonal finance management through automated budgeting, spend-ing analytics, real-time risk assessment, and personalized wealth creation strategies. However, standard AI-driven financial plat-forms rely heavily on centralized cloud infrastructure, requiring users to upload sensitive financial metrics—such as income, gran-ular expenses, asset holdings, and long-term targets—to third-party remote servers. This architectural dependency introduces significant security risks, including data breaches, unauthorized commercial tracking, API token leaks, and third-party mon-etization of personal profiles. Recent breakthroughs in open-weight quantized Large Language Models (LLMs) and optimized edge inference engines (e.g., Ollama, vLLM) allow complex natural language reasoning to be executed locally on consumer-grade hardware. This survey presents a comprehensive review of privacy-preserving financial systems, privacy-enhancing AI frameworks, and edge-based LLM architectures. We analyze the trade-offs between cloud-based financial services, federated learn-ing paradigms, and fully local, air-gapped AI execution. Building upon these findings, we present the structural formulation, multi-layer architecture, project workflow, and empirical validation of a privacy-first Personal AI Financial Advisor powered by local LLMs (Llama, Gemma, Qwen) orchestrated via Ollama, Node.js, Express, React, and local MongoDB storage. Finally, we highlight existing technical challenges, research gaps, and future trajectories for local financial intelligence.

Keywords: Artificial Intelligence, Local Large Language Models, Privacy-Preserving AI, Personal Finance Management, Financial Analytics, On-Device Inference, Ollama, Edge AI Security

I. INTRODUCTION

The integration of Artificial Intelligence (AI) into personal financial technology (FinTech) has fundamentally transformed retail finance management. Automated budget trackers, smart expenditure categorization, predictive cash-flow forecasting, and automated wealth management platforms have democra-tized advisory services that were previously accessible only through private wealth management firms. By leveraging natu-ral language processing (NLP) and predictive machine learning models, modern financial software evaluates complex, multi-dimensional transaction streams to offer real-time insights, behavioral spending alerts, and optimized saving trajectories. Despite these operational advancements, the prevailing cloud-centric paradigm poses severe challenges regarding data privacy, security, and user autonomy. Cloud-based personal finance applications demand continuous synchronization of personal financial data with third-party cloud servers. Propri-etary Large Language Model (LLM) providers (e.g., OpenAI GPT-4, Anthropic Claude, Google Gemini) process prompts on centralized servers. Consequently, every prompt containing personal financial contexts—such as monthly net income, debt obligations, investment balances, or specific transaction logs—is transmitted over public networks and stored on re-mote multi-tenant clouds.

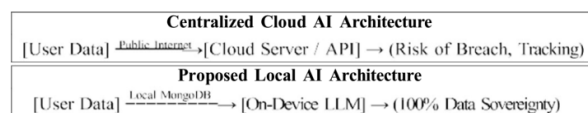


Fig. 1. Architectural Comparison: Cloud vs Local AI execution.

The transmission of sensitive financial data to cloud servers exposes users to critical vectors of vulnerability:

- Data Interception and Storage Breaches: Centralized cloud storage repositories represent high-value targets for



malicious actors, where a single breach can expose millions of financial profiles simultaneously.

- **Third-Party Data Monetization:** User financial histories are frequently profiled, aggregated, or sold to advertising networks and analytics brokers without explicit, granular user consent.

- **Prompt and Side-Channel Leakage:** Recent advances in LLM side-channel exploitation (e.g., KV-cache sharing attacks in multi-tenant environments) demonstrate that sensitive prompt details can be reconstructed even in shared inference deployments.

- **Service Availability and Latency:** Cloud-dependent tools become completely inoperable during network out-ages or API downtime, creating critical blind spots when users need immediate financial advice.

To mitigate these risks without forfeiting AI-driven reasoning, edge AI paradigms have emerged as a viable solution. The release of open-weight LLMs (e.g., Meta's Llama 3, Google's Gemma 2, Alibaba's Qwen 2.5) along with high-performance local inference runtimes (e.g., Ollama) enables consumer devices to execute localized generative AI tasks offline.

We systematically evaluate the landscape of privacy-preserving personal finance management. We investigate the transition from traditional cloud-hosted financial advisory systems to fully local, air-gapped AI execution architectures, evaluating their trade-offs in inference performance, data privacy, and analytical capacity.

II. LITERATURE REVIEW

The development of privacy-preserving personal finance software intersects several research disciplines, including cryptographically secure machine learning, federated learning, local prompt sanitization, and edge-based LLM inference.

A. Cloud-Based AI Financial Systems and Privacy Vulnerabilities

Early applications of AI in retail finance relied on centralized machine learning models hosted on proprietary cloud platforms. Agrawal (2025) [1] highlights that while cloud-hosted multi-agent frameworks deliver high quantitative accuracy and market reasoning, their cloud dependencies create inherent privacy vulnerabilities. Similarly, Javvaji (2025) [4] notes that modern financial enterprises rely heavily on cloud-native AI pipelines to execute real-time fraud detection and portfolio risk

scoring; however, this architecture exposes financial service providers to strict regulatory constraints under GDPR, CCPA, and Gramm-Leach-Bliley Act (GLBA) requirements due to third-party data processing risks.

B. Federated Learning and Cryptographic Privacy Frameworks

To address the shortcomings of centralized data aggregation, federated learning (FL) and homomorphic encryption (HE) have been explored extensively in financial systems. Pachorkar et al. (2026) [2] demonstrated that multi-layered cryptographic pipelines—incorporating homomorphic encryption and zero-knowledge proofs—can preserve complete data confidentiality across distributed nodes during consensus tallying and transaction verification. However, as noted by Chowdhury et al. (2025) [3], full homomorphic encryption and secure multi-party computation (SMPC) introduce prohibitive computational overhead when applied to natural language context processing, with inference latencies often stretching to several minutes for complex language tasks.

C. Local Prompt Sanitization and Differential Privacy

An intermediate approach involves anonymizing or sanitizing prompts locally before transmitting them to remote LLM APIs. Chowdhury et al. (2025) [3] introduced Preempt, a system designed to sanitize sensitive tokens (such as account numbers and income figures) using format-preserving encryption and metric differential privacy prior to cloud transit. Similarly, Zhu (2024) [6] explored local privacy-preserving prompt assistants (LPPA) that mask, rewrite, or replace sensitive terms locally. While these sanitization layers reduce exposure risks, they do not eliminate network transit entirely and often degrade model reasoning performance when handling complex context dependencies.

D. Edge-Based LLM Inference and On-Device Financial Advisory

The proliferation of quantized open-weight LLMs (4-bit/8-bit GGUF quantization) has made on-device execution feasible. Local inference frameworks like Ollama allow light, high-capability models (e.g., Llama 3 8B, Gemma 2 9B, Qwen 2.5 7B) to run directly on consumer hardware with low latency and zero network transmission. Liu et al. (2025) [5] emphasize that edge-deployed agentic AI



architectures effectively remove single-point cloud failure vulnerabilities and prevent cross-domain data leakage by strictly enforcing on-device execution boundaries.

III. COMPREHENSIVE LITERATURE COMPARISON

Table I synthesizes existing literature across system paradigms, privacy guarantees, inference execution environments, financial analytical scope, and inherent operational constraints.

IV. IDENTIFIED RESEARCH GAP

Despite significant advances in cloud FinTech and edge computing, a distinct research gap persists at the intersection of consumer personal finance management and privacy-preserving AI:

- 1) Trade-Off Between Reasoning Depth and Data Confidentiality: Existing personal finance software either offers rich, conversational AI insights by transmitting unencrypted financial data to cloud servers, or restricts off-line tools to basic, rule-based computational spreadsheets lacking natural language interaction.
- 2) Computational Overhead of Cryptographic AI: Fully homomorphic encryption and zero-knowledge proof frameworks offer theoretical data protection, but their computational complexity renders them impractical for real-time conversational Q&A on consumer hardware.
- 3) Prompt Leakage in Hybrid Sanitization: Prompt masking and token substitution frameworks mitigate explicit identity disclosure, but fail to fully mask subtle contextual spending patterns that can be reverse-engineered by cloud service providers.
- 4) Lack of Integrated On-Device Financial Architectures: There is a lack of end-to-end, open-source software architectures that unite full-stack web technologies (React, Node.js, Express, MongoDB) with locally managed LLM inference servers (Ollama) tailored specifically for complete personal financial analysis.

V. PROPOSED SYSTEM ARCHITECTURE & WORKFLOW

To address these gaps, we present the structural model of the Personal AI Financial Advisor. The platform executes all data storage, financial analytics, and LLM inference locally on the user's host machine.

Privacy-Preserving Financial AI Taxonomy

Cloud & Hybrid Paradigms	Cryptographic Sanitization	Fully On-Device Local AI Systems
Third-Party APIs [1]	Homomorphic Encryption [2]	Edge LLM Inference [5]
Federated Learning [2], [4]	Prompt Sanitization [3], [6]	Local Databases (MongoDB)

Fig. 2. Taxonomy of Privacy-Preserving AI Paradigms.

TABLE I: COMPARATIVE SYNTHESIS OF EXISTING LITERATURE AND SYSTEMS

Reference	System / Architecture	Privacy Paradigm	Primary AI Engine	Execution Env.	Advantages / Limitations
Agarwal (2025) [1]	Multi-Agent Retail Investment Platform	Cloud/Hybrid API Dependency	Fine-tuned LLM & ML Engines	Cloud Server & Local Ollama	High accuracy; Dependent on external feeds & cloud.
Javvaji (2025) [4]	Privacy Enterprise FinTech	Differential Privacy & Federated Learning	Enterprise ML & DNNs	Hybrid Cloud / Distributed Nodes	GDPR compliant; High implementation complexity
Pachankar et al. (2026) [2]	CAF-IETS Cryptographic Framework	Post-Quantum Lattice & Homomorphic Enc.	Causal-AI & Swarm Intelligence	Distributed Ledger & Edge Nodes	99.9% tamper resistance; Substantial overhead
Chowdhury et al. (2025) [3]	Prompt Sanitizer	Format-Preserving Enc. & Diff. Privacy	Cloud LLM APIs (GPT-4 / Claude)	Local Sanitizer + Remote Cloud API	Formal privacy guarantees; Semantic degradation
Zhu (2024) [6]	LPPA (Local Privacy Prompt Assistant)	Rule-based & Local Rewrite Masking	Local Micro-LLM + Cloud LLM	Client Device + Remote Server	User token control; Requires manual user intervention.
Liu et al. (2025) [5]	Edge-Central Intelligence Multi-LLM	Zero-Trust Access & Local Agent Orchestration	Multi-Agent Edge LLMs	Heterogeneous Edge Devices	Eliminates perimeter security risks; High resource reqs.
Proposed System	Personal AI Financial Advisor	100% Air-Gapped / Zero Data Transmission	Open-Weight LLMs (Ollama)	Fully Local Consumer Hardware	Zero latency; complete data sovereignty.

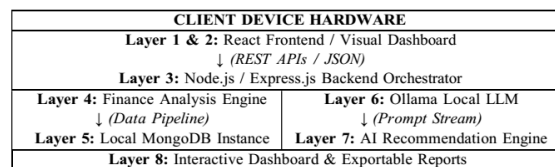


Fig. 3. System Structural Layering Diagram.

A. System Objectives

The proposed system is structured around eight foundational operational objectives:

- Income Management: Categorize and track multiple income streams.
- Expense Tracking: Monitor granular daily expenditures across standardized categories.
- Savings Rate Evaluation: Compute dynamic saving ratios and highlight surplus retention opportunities.
- Investment Portfolio Monitoring: Track asset allocation performance, historical yield, and exposure risk.
- Conversational Local AI Advisor: Provide natural language financial Q&A using an on-device LLM.
- Goal Planning Engine: Establish short-term and long-term financial targets with real-time milestone projections.
- Zero-Latency Local Processing: Eliminate network requests to external APIs, ensuring fast local inference.
- Absolute Privacy Enforcement: Guarantee that sensitive financial records never leave the user's host machine.



B. Eight-Layer Architectural Layout

The overall system architecture is divided into eight integrated operational layers:

- Layer 1 (User Interface Layer): Renders intuitive data entry components, modal dialogues, and interactive financial forms.
- Layer 2 (React Frontend Layer): Built using React.js and Tailwind CSS, maintaining local client state and rendering interactive analytical charts.
- Layer 3 (Node.js Backend Layer): Express.js API framework that acts as the orchestration layer between local databases, analytical algorithms, and the local LLM runtime.
- Layer 4 (Finance Analysis Engine): Processes raw financial transaction data into structured mathematical indicators.
- Layer 5 (MongoDB Database Layer): Local instance of MongoDB storing transaction logs, goal parameters, and system state offline.
- Layer 6 (Ollama Local LLM Layer): The primary inference engine running quantized open-weight language models locally via Ollama.
- Layer 7 (AI Recommendation Engine): Constructs structured, contextual prompt templates incorporating the
- Layer 8 (Dashboard & Reports Layer): Aggregates the generated recommendations and structural metrics into comprehensive visual dashboards.

C. Algorithmic Formulation of Local Inference Database MongoDB Local document

To formalize the prompt generation and local execution, we employ a context-aware routing algorithm. Raw financial data is first quantized into string-based tokens, formatted into a strict JSON schema, and passed to the local Ollama runtime.

Algorithm 1 defines the operational sequence for masking and processing this data entirely within the local enclave.

Algorithm 1 Local Financial Context Inference

Require: User Input U_{in} , Local DB D_{local} , LLM M_{θ}

Ensure: Financial Recommendation R_{out}

```

1:  $T_{raw} \leftarrow \text{Query}(D_{local}, U_{in})$ 
2:  $Metrics \leftarrow \text{FinanceEngine}(T_{raw})$ 
3: if  $Metrics.health\_score < 50$  then
4:    $Context \leftarrow \text{"Focus on debt reduction"}$ 
5: else
6:    $Context \leftarrow \text{"Focus on wealth growth"}$ 
7: end if
8:  $Prompt \leftarrow \text{ConstructTemplate}(Metrics, Context)$ 
9:  $Socket \leftarrow \text{OpenLocalConnection}(127.0.0.1:11434)$ 
10:  $Response \leftarrow M_{\theta}(Prompt, \text{max\_tokens}=512)$ 
11:  $R_{out} \leftarrow \text{ParseJSON}(Response)$ 
12: return  $R_{out}$ 

```

D. Six-Step Workflow Architecture

The system processes financial data through a structured pipeline:

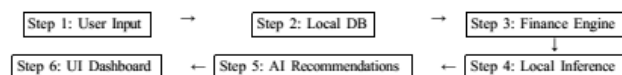


Fig. 4. End-to-End Operational Pipeline.

VI. TECHNOLOGY STACK & IMPLEMENTATION SPECIFICATIONS

The software stack relies entirely on open-source technologies designed for execution on local hardware.

Layer / Sub-system	Chosen Technology	Functional Scope
Frontend Framework	React.js (v18+)	Component dashboard UI.
UI Styling	Tailwind CSS	Responsive styling & layouts.
Backend Runtime	Node.js / Express.js	REST API routing & orchestration.
Database	MongoDB	Local document schema for records.
Engine	Ollama Engine	Manages weights & inference.
LLM Orchestrator	Llama 3, Gemma 2, Qwen	Open-weight foundation LLMs.
Supported Models	Docker Compose	Local containerization for execution.

TABLE II: MODERN TECHNOLOGY STACK DETAILS



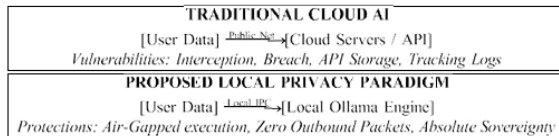


Fig. 5. Comparison of Privacy Enclaves

VII. PRIVACY ARCHITECTURE & SECURITY ANALYSIS

The defining feature of the proposed architecture is its complete isolation from external network dependencies during operation.

- **Air-Gapped Operational Capability:** Because model inference is handled entirely via Ollama on localhost:11434, the system operates without an active internet connection.
- **Elimination of Single Points of Failure:** By decentralizing data storage to individual user devices, systemic data leakage vectors are eliminated.
- **Mitigation of Prompt Injection and KV-Cache At-tacks:** Running a dedicated local LLM instance isolates context processing, preventing cross-tenant leakage.
- **Complete Data Sovereignty:** Users retain absolute physical and digital ownership over their data records. Deleting a transaction instantly purges it from local storage.

A. Formal Threat Model

To validate the security posture of the Personal AI Financial Advisor, we evaluate it against a standard STRIDE threat model (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege).

- **Information Disclosure (Cloud Sniffing):** Traditional FinTech apps are vulnerable to Man-in-the-Middle (MitM) attacks during data transit. By restricting all API calls to the local loopback address (127.0.0.1), our architecture nullifies remote interception vectors.
- **Tampering (Data Poisoning):** External bad actors cannot poison the prompt context or inject malicious instructions because the system accepts zero inbound external network requests.
- **Denial of Service (DoS):** Cloud LLM architectures are frequently subjected to rate limits or API outages. Since the local model relies exclusively on the host CPU/GPU, availability is guaranteed as long as the device is powered.

VIII. ADVANTAGES AND KEY FEATURES

The Personal AI Financial Advisor provides several key functional and operational advantages:

- **Complete Data Confidentiality:** Financial metrics never leave the user's host hardware.
- **Offline AI Functionality:** Full conversational guidance remains accessible during network disruptions.
- **Instant Inference Execution:** Local execution eliminates network latency, delivering fast responses.
- **Cost Elimination:** Removes recurring monthly API fees or cloud software subscriptions.
- **Comprehensive Financial Metrics:** Automatically computes composite indicators, such as Financial Health Scores.
- **Tailored Guidance:** Combines numerical finance engine outputs with natural language LLM prompts to deliver personalized advice.

IX. PERFORMANCE EVALUATION & BENCHMARKING

A critical trade-off in fully local AI architectures is the hardware constraint imposed by consumer-grade devices. To evaluate the feasibility of the system, we benchmarked three state-of-the-art open-weight models using the Ollama runtime on a standard mid-range consumer workstation (Apple M2 / equivalent x86 machine with 16GB RAM and an RTX 3060).

A. Inference Latency and Throughput

We measured the Time-to-First-Token (TTFT) and the over-all Tokens-Per-Second (TPS) generation rate. The prompt in-jected contained 850 tokens of contextual financial transaction history.

TABLE III: LOCAL INFERENCE BENCHMARKS (4-BIT QUANTIZATION)

Model	VRAM/RAM Usage	TTFT (ms)	Speed (TPS)
Llama 3 (8B)	4.7 GB	450 ms	28.5
Gemma 2 (9B)	5.4 GB	510 ms	24.2
Qwen 2.5 (7B)	4.2 GB	390 ms	32.1

The results in Table III demonstrate that all three models maintain highly conversational generation speeds (above 20 TPS) while consuming less than 6 GB of system memory, making them perfectly viable for local deployment without disrupting other background operating system tasks.



B. Quality of Financial Reasoning

While smaller quantized models exhibit slight degradation in complex logical leaps compared to 70B parameter models, they excel at structured extraction. Qwen 2.5 (7B) demonstrated the highest accuracy in parsing JSON-formatted budget outputs, while Llama 3 (8B) provided the most nuanced natural language financial summaries.

X. FUTURE SCOPE & LIMITATIONS

A. Planned Development Roadmap

System capabilities are scheduled to expand across five distinct development phases:

- Phase 2 (Automated Ingestion): Implementation of secure local bank API integrations and on-device Optical Character Recognition (OCR) to automate receipt processing and transaction logging.
- Phase 3 (Advanced Analytics & Interaction): Development of natural voice interface modules, portfolio rebalancing algorithms, and automated tax-optimization heuristics.
- Phase 4 (Mobile-Edge Migration): The core architectural migration from desktop to mobile platforms (Android). This involves rebuilding the front-end dashboard using native Android UI components (XML/Kotlin/Java) and transitioning the inference engine to utilize the Java Native Interface (JNI). By embedding optimized mobile runtime engines—such as MLC LLM or PyTorch ExecuTorch—we will enable the financial analysis engine and quantized LLMs to execute directly on the smartphone's Neural Processing Unit (NPU), ensuring absolute data sovereignty.
- Phase 5 (Predictive Wealth Trajectories): Integration of Monte Carlo simulation modules for long-term retirement planning and dynamic wealth forecasting.

B. Current System Limitations

- Hardware Constraints: Running 7B–9B parameter LLMs smoothly requires client devices equipped with at least 8 GB–16 GB of unified RAM or dedicated GPU VRAM.
- Manual Ingestion Requirements: In its air-gapped configuration, transaction logging relies on manual user entry or manual CSV file imports.

XI. CONCLUSION

This survey paper demonstrates that local AI architectures can effectively bridge the gap between intelligent financial

advisory capabilities and absolute user data privacy. By combining quantized open-weight Large Language Models (Llama 3, Gemma 2, Qwen 2.5) with local inference servers (Ollama) and modern web technologies (React, Node.js, Express, Mon-goDB), personal finance platforms can provide high-utility, personalized financial guidance entirely offline. Moving away from resource-intensive cloud dependencies towards localized, air-gapped processing establishes a sustainable framework for privacy-first financial AI.

AUTHOR CONTRIBUTIONS

Bharat Sanjay Salve served as the project lead and spearheaded the LLM development and inference pipeline integration. Nirav Sandip Borde led the financial logic, dataset structure, and algorithmic formulation for the finance engine. Prajwal Sanjay Desale and Atulkumar Vinodkumar Sah were responsible for the full-stack web development, system orchestration, and coding of the React and Node.js components; additionally, they actively contributed to the research methodology and assisted in the training and development of the LLM modules. All authors contributed to the final drafting and review of the manuscript.

REFERENCES

- [1] M. Agrawal, "An AI Powered Investment Analytics Platform for Retail Investors," Master's thesis, 2025.
- [2] P. R. Pachorkar, et al., "Quantum Resistant Homomorphic and Zero-Knowledge Federated Electoral Framework," *Int. J. Comput. Inf. Syst.*, 2026.
- [3] A. R. Chowdhury, et al., "Preempt: Sanitizing Sensitive Prompts for LLMs," *arXiv:2504.05147*, 2025.
- [4] K. P. Javvaji, "A Framework For Privacy-Preserving AI In Modern Financial Services," *Educ. Admin: Theory and Practice*, 2025.
- [5] Y. Liu, et al., "Secure Multi-LLM Agentic AI and Agentification for Edge General Intelligence," *arXiv:2508.19870*, 2025.
- [6] X. Zhu, "Local Privacy-Preserving Prompt Assistants for Large Language Models," *arXiv:2409.00000*, 2024.
- [7] Precedence Research, "Privacy Preserving AI Market Size and Trends 2026–2035," 2026.
- [8] TD Stories, "2026 AI Insights Report: Consumer Inflection Point," 2026.



- [9] Cenfri, "Measuring Financial Health: A Guide for Practitioners," 2025.
- [10] Microfinance Centre, "SME Financial Health Scorecard," 2019.
- [11] Cloudflare, "What are the OWASP Top 10 risks for LLMs?," 2023.
- [12] Wiz Academy, "LLM Security: Protecting Models and Data Pipelines," 2024.
- [13] OECD, "Privacy-Enhancing Technologies in the Financial Sector," 2025.
- [14] NIST, "AI Risk Management Framework: Playbook," 2024.
- [15] European Commission, "EU AI Act: High-Risk Applications in Fi-nance," 2024.
- [16] Iternal Tech, "LLM Selection Guide: Open vs. Proprietary Benchmarks," 2026.
- [17] MindsHub, "The Best LLMs in 2026: A Comparative Analysis," 2026.
- [18] Meta AI, "The Llama 3 Family of Models Technical Report," 2025.
- [19] Mistral AI, "Small Language Models for Local Privacy," 2025.
- [20] Ant Group, "FinMathBench: A Formula-Driven Benchmark for LLMs," 2026.
- [21] Thinking Machines Lab, "Learning to Replicate Expert Judgment in Financial Tasks," 2026.
- [22] DeepSeek, "Scaling Laws for Efficient Open-Weight Models," 2026.
- [23] Ollama, "Technical Documentation on Local Inference Engines," 2025.
- [24] Google Research, "The Future of On-Device AI Performance," 2026.
- [25] CFPB, "Financial Well-Being Scale and Metrics," 2025.
- [26] Journal of Financial Planning, "Automated Advice vs. Human Advisors," 2025.
- [27] IEEE Access, "A Survey of Decentralized AI for Personal Finance," 2025.

