Preprints.org

Review

Not peer-reviewed version

A Comprehensive Survey of AI Agent Frameworks and Their Applications in Financial Services

Satyadhar Joshi *

Posted Date: 13 May 2025

doi: 10.20944/preprints202505.0971.v1

Keywords: AI Agents; Agentic AI; Financial Services; Multi-Agent Systems; Generative AI; Risk Management; multi-agent systems; financial technology; LLMs; autonomous agents; frameworks

Review

A Comprehensive Survey of AI Agent Frameworks and Their Applications in Financial Services

Satyadhar Joshi

Independent, Alumnus, International MBA, Bar-Ilan University, Israel; satyadhar.joshi@gmail.com

Abstract: This paper surveys the landscape of AI agent frameworks, highlights their core features and differences, and explores their applications in financial services. We synthesize insights from recent industry reports, academic research, and technical blog posts, focusing on frameworks such as CrewAI, LangGraph, LlamaIndex, and others. We also discuss the challenges and opportunities of deploying agentic AI in production environments, with an emphasis on financial trading, investment analysis, and decision support. We analyze the rapidly evolving landscape of agentic AI systems, focusing on their architecture, capabilities, and practical implementations in banking, trading, and risk management. The study examines prominent frameworks including LangGraph for stateful agent orchestration, CrewAI for collaborative multi-agent workflows, and AutoGen for conversational agent systems, alongside industry platforms like IBM watsonx and NVIDIA NIM. This paper presents a comprehensive survey of AI agent frameworks and their transformative impact on the financial services industry. The study examines both technical frameworks (LangGraph, CrewAI, AutoGen, etc.) and practical implementations in financial institutions. We highlight productivity gains (up to 80% time reduction in data tasks), risk management improvements, and workforce transformation challenges. The paper concludes with recommendations for financial institutions adopting agentic AI solutions. Our analysis reveals three key findings: (1) specialized agent frameworks achieve 50-80% productivity gains in financial data tasks compared to traditional approaches, (2) multi-agent systems demonstrate particular promise in complex domains like algorithmic trading and fraud detection, and (3) successful deployment requires addressing critical challenges in workforce upskilling, risk alignment, and regulatory compliance. The paper provides a theoretical foundation for agentic AI in finance, introducing formal models for agent design patterns, multimodal fusion, and market microfoundations. We further present a summary of several evaluation frameworks for assessing agent performance across financial use cases, including portfolio optimization and AML compliance. The study concludes with recommendations for financial institutions adopting agentic AI, emphasizing the need for standardized architectures, robust testing protocols, and hybrid human-AI workflows.

Keywords: AI Agents; Agentic AI; Financial Services; Multi-Agent Systems; Generative AI; Risk Management; multi-agent systems; financial technology; LLMs; autonomous agents, frameworks

---

1. Introduction

The emergence of agentic AI—AI systems capable of autonomous, multi-step reasoning and action—marks a new era in artificial intelligence [1–3]. Large Language Models (LLMs) have enabled agents to reason, plan, and interact with complex environments, making them suitable for a wide range of enterprise and industrial applications [4]. In the financial sector, agentic AI is being explored for tasks ranging from trading and investment analysis to compliance and workflow automation [5–7].

The financial services industry is undergoing a radical transformation through the adoption of AI agent frameworks [8]. As noted by [4], generative AI is becoming a utility similar to electricity, with multi-agent systems emerging as the next evolutionary step. Recent developments in 2024-2025 show remarkable progress in both agent frameworks and their financial applications [9–11].

However, several important considerations need to be addressed in future work:

- **Scalability**: Evaluating the scalability of the architecture to handle a large number of agents and complex financial scenarios.
- **Real-time Performance**: Assessing the system's ability to operate in real-time and make timely decisions in dynamic market conditions.
- **Regulatory Compliance**: Ensuring that the system complies with relevant financial regulations and ethical guidelines.
- **Integration with Existing Systems**: Exploring how the architecture can be integrated with existing financial systems and infrastructure.

1.1. Related Work

The intersection of multi-agent systems and artificial intelligence has a long history, with early research focusing on distributed problem-solving, cooperative robotics, and game theory. Chen [12] laid the groundwork for applying computationally intelligent agents in economics and finance, demonstrating their ability to model complex market behaviors and agent interactions.

The advent of Large Language Models (LLMs) has revolutionized the field of AI, enabling the development of more sophisticated and autonomous agents. Agentic AI, as highlighted by Pounds [1] and Jadhav [2], represents a paradigm shift towards AI systems that can reason, plan, and act autonomously, significantly expanding the potential of MAS. Winston [3] emphasizes the importance of understanding AI agents and their growing impact.

Several platforms and frameworks have emerged to facilitate the development of AI agents. LangChain [13] provides a versatile toolkit for building agents that can interact with external data sources and tools. LangGraph [14] offers a lower-level abstraction for building stateful and interactive agentic applications. CrewAI [15] focuses on orchestrating collaborative multi-agent workflows. AutoGen [16] simplifies the creation of multi-agent conversations. Other notable frameworks include Semantic Kernel [17], Agentforce [18], Mosaic AI Agent Framework [19,20], and platforms offered by major cloud providers such as Google Cloud's Vertex AI Agent Builder [21], Amazon Bedrock Agents [22], Azure Cosmos DB [23], and IBM watsonx.ai [24]. Pydantic-AI [25,26] provides tools for integrating Pydantic with LLMs in agent development.

Comparative analyses of these frameworks, such as those by Aydın [27], Relari AI [10], and others [9,11,28–31], offer valuable insights into their strengths, weaknesses, and suitability for different applications.

The application of AI agents in the financial domain is a rapidly evolving area. Reports from McKinsey [8] and the World Economic Forum [32] highlight the transformative potential of agentic AI in revolutionizing financial services. Specific applications include AI traders in financial markets [5,33], LLM-based multi-agent systems for financial decision-making [7], and open-source AI agent platforms for financial applications [34]. Research is also exploring the use of AI agents to enhance investment analysis [6] and improve employee productivity in financial institutions [35,36]. Cognizant [37] and other companies are developing AI solutions for the financial sector.

However, the adoption of AI agents in finance also raises significant concerns. Risk management is paramount [38–41], and ensuring the responsible and ethical use of AI is crucial [42,43]. The Financial Stability Board [44] and central banks like the European Central Bank [45] are actively addressing the potential risks associated with AI in financial services. Moody's Analytics has also explored the rise of AI agents in finance [46,47]. International Banker also discusses balancing risk and workforce transformation [48].

1.2. Overview of Modern AI Agent Frameworks

A variety of frameworks have emerged to support the development and deployment of AI agents. These range from open-source libraries to enterprise-grade platforms.

 doi:10.20944/preprints202505.0971.v1

1.2.1. General-Purpose Frameworks

LangGraph is a low-level orchestration framework for building controllable agents with state management and debugging tools [14]. CrewAI specializes in collaborative, role-based agent teams [15]. LlamaIndex (llama-agents) focuses on connecting LLMs to enterprise data for knowledge-intensive applications [49]. Other notable frameworks include PydanticAI [25,26], Semantic Kernel [17], and AutoGen [16].

1.2.2. Industry and Cloud Solutions

Major cloud providers and enterprise vendors have launched agentic AI platforms such as NVIDIA NIM [50], IBM watsonx [24,51], Amazon Bedrock Agents [22], and Salesforce Agentforce [18]. These platforms offer integration with business APIs, scalability, and compliance features.

1.2.3. Comparative Analyses

Recent comparative studies and blog posts provide overviews of the most popular frameworks, their architectures, and use cases [9–11,27,28,30,31].

2. AI Agent Frameworks: A Comparative Analysis

The development of effective multi-agent systems relies heavily on the underlying AI agent frameworks. This section provides a comparative analysis of several prominent frameworks, highlighting their key features, strengths, and weaknesses.

2.1. LangChain

LangChain [13] is a versatile framework that simplifies the integration of LLMs with external data sources and tools. Its modular design allows developers to create agents with diverse capabilities, including information retrieval, code execution, and web browsing. LangChain's strength lies in its flexibility and extensive ecosystem of integrations.

2.2. LangGraph

LangGraph [14] provides a lower-level abstraction for building stateful and interactive agentic applications. It enables the creation of complex agent workflows with explicit control over agent interactions and state transitions. LangGraph is particularly suitable for applications that require fine-grained control over agent behavior.

2.3. CrewAI

CrewAI [15] focuses on orchestrating collaborative multi-agent workflows. It allows developers to define agents with specific roles and responsibilities and to coordinate their interactions to achieve complex tasks. CrewAI is well-suited for applications that involve teamwork and division of labor among agents.

2.4. AutoGen

AutoGen [16] simplifies the development of multi-agent conversations. It enables the creation of agents that can communicate with each other to solve problems collaboratively. AutoGen is particularly useful for building conversational AI systems and applications that require complex reasoning and debate.

2.5. Other Frameworks

Other notable frameworks include Semantic Kernel [17], which emphasizes the integration of semantic functions with LLMs, and platforms offered by cloud providers such as Google Cloud's Vertex AI Agent Builder [21] and Amazon Bedrock Agents [22], which provide tools for building and deploying agents within their respective cloud ecosystems. IBM watsonx.ai [24] also offers agent development capabilities. Agentforce [18] and Mosaic AI Agent Framework [19] are also noteworthy.

2.6. *Comparison and Suitability for Finance*

The choice of an appropriate agent framework depends on the specific requirements of the financial application. For example, LangChain's flexibility might be suitable for building agents that need to access diverse financial data sources, while CrewAI could be beneficial for developing systems that involve teams of agents performing different analytical tasks. AutoGen might be used for sophisticated financial forecasting. Factors such as scalability, robustness, explainability, and the availability of specific financial tools and libraries should also be considered.

3. Proposed Multi-Agent Architecture

This section presents our novel multi-agent architecture for advanced financial analysis. Our architecture is designed to leverage the strengths of LLMs.

3.1. *Architecture Overview*

Our architecture comprises three key layers:

- **Data Layer**: This layer is responsible for collecting, storing, and managing financial data from various sources. It includes components for data acquisition, preprocessing, and storage. Technologies like Retrieval Augmented Generation (RAG) [52] can be employed to enhance the agent's ability to access and utilize relevant information from this layer. Data pipelines using LLMs and multi-agent systems are discussed in [4].
- **Agent Layer**: This layer consists of a collection of intelligent agents, each specialized in a specific financial task. Agents in this layer are powered by LLMs and equipped with capabilities for reasoning, planning, communication, and action execution.
- **Orchestration Layer**: This layer is responsible for coordinating the activities of agents in the Agent Layer. It manages agent communication, task allocation, conflict resolution, and overall system behavior. Frameworks like Camel [53] can provide useful insights into designing effective communication protocols.

3.2. *Agent Design*

Each agent in the Agent Layer is designed with the following components:

- **LLM-Powered Cognition Module**: This module utilizes a pre-trained LLM, fine-tuned on financial data, to perform tasks such as data analysis, forecasting, risk assessment, and report generation.
- **Domain Knowledge Base**: This module stores domain-specific knowledge, including financial concepts, market regulations, and company information.
- **Communication Interface**: This module enables agents to communicate with each other and with the Orchestration Layer using a standardized message format.
- **Action Execution Engine**: This module executes the actions determined by the agent's cognition module, such as retrieving data, performing calculations, and generating reports. Pydantic [26] can be used to ensure data integrity.

3.3. *Orchestration Mechanisms*

The Orchestration Layer employs a combination of techniques to manage agent interactions:

- **Task Decomposition**: Complex financial tasks are decomposed into smaller subtasks that can be assigned to individual agents.
- **Agent Negotiation**: Agents negotiate with each other to determine the best way to execute their assigned tasks.
- **Conflict Resolution**: Mechanisms are in place to resolve conflicts that may arise between agents.
- **System Monitoring**: The Orchestration Layer monitors the overall system performance and intervenes when necessary.

4. Evaluation Strategy

To rigorously evaluate the effectiveness of our proposed multi-agent architecture, we define a comprehensive evaluation strategy.

4.1. Evaluation Scenarios

We will evaluate our architecture in the following financial scenarios:

- **Portfolio Optimization**: Agents will collaborate to optimize investment portfolios based on risk tolerance, return objectives, and market conditions.
- **Fraud Detection**: Agents will analyze transaction data to identify patterns indicative of fraudulent activity. AI agents are being developed to fight financial crime [54].
- **Algorithmic Trading**: Agents will develop and execute trading strategies in a simulated market environment [33].
- **Financial News Analysis**: Agents will analyze news articles and social media data to identify market trends and sentiment.

4.2. Evaluation Metrics

We will evaluate the performance of our architecture using the following key performance indicators (KPIs):

- **Accuracy**: The accuracy of agent predictions and decisions in each scenario.
- **Efficiency**: The speed and resource consumption of the system in completing tasks.
- **Robustness**: The ability of the system to handle noisy or incomplete data and unexpected events.
- **Explainability**: The degree to which agent decisions can be explained and justified.
- **Risk-Adjusted Return**: A measure of investment performance that considers the level of risk taken.

4.3. Benchmarking and Baselines

We will compare the performance of our architecture against the following baseline methods:

- **Baseline 1**: A traditional rule-based system that uses predefined rules to perform financial analysis.
- **Baseline 2**: A single-agent system that utilizes an LLM but does not involve multi-agent coordination.
- **Baseline 3**: Existing state-of-the-art financial models (where applicable to the scenario).

5. Agentic AI in Finance

The finance sector is at the forefront of adopting agentic AI due to its need for automation, data analysis, and risk management.

5.1. Risk Management

Agentic AI shows particular promise in financial risk:

- 45% of firms now use GenAI for risk management [39]
- Credit risk analysis improvements through agent collaboration [40]
- Automated AML/KYC processes via specialized agents [54]

5.2. Trading and Investment

Multi-agent systems are transforming trading:

- [33] demonstrate AI trader impact on markets
- FinRobot [34] provides open-source platform for financial LLMs
- Multimodal agents combine diverse data sources [5]

 doi:10.20944/preprints202505.0971.v1

5.3. Productivity Enhancements

- Capitec Bank reports 1+ hour weekly savings per employee [55]
- West Monroe's agent reduces data task time by 80% [56]
- JPMorgan's AI assistant improves operations [57]

5.4. Customer Experience

- Interface.ai's agentic copilot boosts efficiency [35]
- Zetaris introduces specialized agents for financial services [58]
- Retrieval-Augmented Generation (RAG) enhances banking services [52]

5.5. Financial Trading and Investment

Researchers have demonstrated the use of multi-agent systems for market modeling and trading [5,33,34]. For example, FinRobot is an open-source agent platform for financial applications using LLMs [34]. Multimodal agents can leverage diverse data sources, tools, and reasoning strategies to optimize trading decisions [5]. Enhanced agent collaboration has been shown to improve investment analysis and financial research outcomes [6].

5.6. Decision Support and Workflow Automation

Agentic AI frameworks are being used to automate data pipelines, compliance checks, and customer support in banking and fintech [4,24,51,59]. Synthesized multi-agent systems can enhance financial decision-making through conceptual reinforcement and collaborative reasoning [7].

5.7. Technical and Safety Considerations

As agentic systems become more autonomous, documenting their technical and safety features is essential [60]. Frameworks like LangGraph and CrewAI offer debugging and state management tools to address these needs [14,15].

6. AI Agent Framework Landscape

The AI agent ecosystem has exploded with numerous frameworks offering distinct capabilities:

6.1. General Purpose Frameworks

- **LangGraph**: A low-level orchestration framework from LangChain enabling controllable agents with state management [14]
- **CrewAI**: Specializes in role-based agent collaboration with built-in task delegation [15]
- **AutoGen**: Microsoft's framework for building multi-agent systems with diverse capabilities [16]
- **Llama-agents**: LlamaIndex's production-ready framework for enterprise knowledge systems [49]
- **Semantic Kernel**: Microsoft's experimental agent framework integrating with AI services [17]

6.2. Industry-Specific Solutions

Financial institutions are adopting specialized platforms:

- NVIDIA NIM for generative AI deployment [50]
- IBM watsonx.ai for enterprise-grade AI development [51]
- Salesforce Agentforce for CRM automation [18]
- AWS Bedrock Agents for business task automation [22]

Recent comparative studies [10,11,31] highlight the strengths of different frameworks. [9] identifies seven top frameworks for 2025, while [28] focuses on multi-agent applications. The Pydantic-AI framework [26] offers unique integration with Python type systems.

 doi:10.20944/preprints202505.0971.v1

7. Cloud Python Libraries for AI Agent Development

Cloud-native Python libraries are central to the rapid development and deployment of agentic AI solutions in finance. These libraries enable scalable, distributed, and production-ready workflows, supporting both experimentation and enterprise applications. The development of AI agents for financial services has been significantly accelerated by cloud-based Python libraries that provide scalable infrastructure and pre-built components. These libraries enable rapid deployment of agentic systems while handling the complexities of distributed computing and cloud integration.

Cloud computing platforms provide a wide array of Python libraries that facilitate the development and deployment of AI agents. These libraries offer functionalities ranging from data storage and retrieval to model training and deployment. Here are some notable examples:

- **Google Cloud Libraries**: Google Cloud offers libraries like Vertex AI, which provides tools to build, deploy, and scale machine learning (ML) models. Vertex AI Agent Builder [21] allows for creating virtual AI agents.
- **Amazon Web Services (AWS) Libraries**: AWS provides services like Amazon Bedrock, and Bedrock Agents [22] which enables the building of generative AI applications.
- **Microsoft Azure Libraries**: Microsoft Azure offers Azure Cosmos DB [23], a database service that can be used to build AI agent memory systems. Additionally, Microsoft's Semantic Kernel [17] can be used in conjunction with Azure services.
- **IBM Cloud Libraries**: IBM Cloud provides watsonx.ai [24,51], a platform with tools for the AI development lifecycle.

These cloud-based Python libraries provide developers with the necessary tools to build and deploy scalable and robust AI agent systems.

7.1. Major Cloud Python Libraries

- **LangChain/LangGraph**: Provides comprehensive tools for building LLM-powered agents with cloud deployment capabilities [14]. The framework supports AWS, GCP, and Azure integration for scalable agent systems.
- **Pydantic-AI**: Offers cloud-optimized agent development with strong typing and validation, particularly useful for financial data pipelines [26]. The library includes connectors for major cloud platforms.
- **IBM watsonx**: Delivers enterprise-grade AI agents with native cloud support through Python SDKs [51]. The platform specializes in secure financial applications with built-in compliance features.
- **Mosaic AI Agent Framework**: Databricks' solution for building autonomous AI assistants with cloud-native architecture [19]. It integrates seamlessly with Databricks' Lakehouse platform for financial data processing.

7.2. Cloud-Specific Implementations

Table 1 summarizes key AI agent libraries across major cloud platforms.

Table 1. Cloud-Specific AI Agent Libraries.

Cloud Platform	Library	Key Feature
AWS	Bedrock Agents	API integration for financial systems [22]
Azure	Semantic Kernel	.NET/Python hybrid agents [17]
GCP	Vertex AI Agent Builder	Financial recommendation systems [21]
Multi-cloud	Camel-AI	Multi-agent coordination [53]

7.3. Financial Services Specialization

Recent advancements in cloud Python libraries specifically target financial applications:

 doi:10.20944/preprints202505.0971.v1

- **FinRobot** [34]: Open-source platform with cloud connectors for market data feeds and trading APIs.
- **Zetaris Agentic AI** [58]: Cloud-native solution for financial data virtualization and agent-based analytics.
- **WorkFusion AI Agents** [54]: Specialized cloud library for anti-financial crime applications with pre-built AML/KYC workflows.

7.4. Performance Considerations

Cloud-based agent systems demonstrate significant performance advantages:

- **Scalability**: Multi-agent systems like those built with [15] can automatically scale across cloud regions during market hours.
- **Latency**: Frameworks such as [10] optimize cloud deployment for low-latency trading applications.
- **Cost Efficiency**: [9] reports cloud-based agents can reduce infrastructure costs by 30-40% compared to on-premise solutions for equivalent workloads.

The evolution of these cloud Python libraries has lowered the barrier to entry for financial institutions adopting agentic AI, while providing the security and compliance features required in regulated environments [61].

7.5. Agno: Cloud-Native Agent Framework

Agno is a Python framework designed for building and deploying LLM-powered agents in the cloud, with features for multi-agent orchestration, cloud deployment, and integration with major providers such as AWS [28]. Agno supports both local and cloud workflows, offering a built-in agent UI, session management, and monitoring tools. Its modular design allows users to connect to models from OpenAI, Anthropic, Cohere, and more, making it suitable for both research and production environments.

7.6. Best Practices for Cloud Python Environments

When deploying agentic systems in the cloud, it is recommended to use isolated Python environments, such as `venv`, to manage dependencies and ensure reproducibility. Agno provides templates and pre-configured codebases to accelerate the transition from prototype to production, with support for monitoring and debugging in distributed cloud settings.

7.7. Alternative Libraries and Approaches

Several other frameworks and libraries also support cloud-based agentic workflows. For example, the PydanticAI project demonstrates how Python type systems can be leveraged for agent orchestration, and offers cloud deployment options [25]. Additionally, the open-source ecosystem continues to expand, with projects like CrewAI and LlamaIndex providing modular, cloud-compatible solutions for multi-agent systems and enterprise data integration [15,49].

7.8. Summary

The trend in cloud Python libraries is toward modularity, composability, and seamless integration with cloud infrastructure. Frameworks like Agno and CrewAI exemplify these principles, enabling the rapid development and deployment of robust agentic AI systems in finance and beyond [15].

8. Theoretical Foundations of Agentic AI

Based on the surveyed literature, we identify ten core theoretical concepts that underpin modern agentic AI systems:

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 May 2025 doi:10.20944/preprints202505.0971.v1

1. **Agentic Design Patterns** - Architectural templates for creating autonomous agents capable of iterative planning and tool use [1]. Characterized by:
$$A = (S, \Pi, M, T) \tag{1}$$
where S=states, Π=policies, M=memory, T=tools.
2. **Multi-Agent Scaling Laws** - Quantitative relationships between agent count and system performance [53]. Demonstrated through:
$$\mathcal{P}(n) \sim n^{\alpha} \log(n) \tag{2}$$
where α is task-dependent.
3. **Verbal Reinforcement Learning** - Conceptual reinforcement through language feedback rather than numeric rewards [7]. Formalized as:
$$\pi_{t+1} = \pi_t + \eta \nabla \mathbb{E}[f_{\text{lang}}(R)] \tag{3}$$
4. **Financial Market Microfoundations** - Agent-based models explaining macro phenomena through individual agent behaviors [33]. Price formation follows:
$$p_{t+1} = p_t + \sum_i w_i a_i(p_t, x_i) \tag{4}$$
5. **Multimodal Fusion Theory** - Framework for combining diverse financial data modalities [5]. Uses attention mechanisms:
$$h = \text{softmax}(QK^T/\sqrt{d})V \tag{5}$$
6. **Agentic Workflow Optimization** - Mathematical formulation of task decomposition in financial processes [6]. Minimizes:
$$\mathcal{L} = \sum_{k=1}^{K} \|T_k - \bigcup_i \pi_i^k\|^2 \tag{6}$$
7. **Conceptual Alignment** - Ensuring agent reasoning aligns with financial domain concepts [7]. Measured by:
$$\mathcal{A} = \mathbb{E}[\text{sim}(c_{\text{human}}, c_{\text{agent}})] \tag{7}$$
8. **Risk-Aware Learning** - Adaptation mechanisms considering financial risk constraints [40]. Policies satisfy:
$$\pi \in \{\pi' | \mathbb{P}(r_{\text{risk}} > \theta) < \epsilon\} \tag{8}$$
9. **Computational Principal-Agent Theory** - Formalizing delegation in AI-human teams [12]. Models:
$$\max_{a \in A} u_p(a) \text{ s.t. } a \in \arg\max_{a'} u_a(a') \tag{9}$$
10. **Generative Economic Equilibrium** - Stable states in AI-augmented financial systems [33]. Requires:
$$\forall i, \pi_i^* \in \text{BR}(\pi_{-i}^*) \tag{10}$$
where BR denotes best response.

These theoretical constructs provide the mathematical foundation for current agentic AI systems in finance, spanning individual agent design to market-scale interactions. The field continues to evolve through formalization of these concepts [44,61].

9. Multi-Agent System Architectures

9.0.1. Microfoundations Market Model

[33] proposes a multi-agent market simulation framework where each agent $a_i \in A$ is modeled as:

$$a_i = (s_i, \pi_i, \theta_i) \tag{11}$$

where:

- s_i: Agent state (e.g., portfolio, risk tolerance),
- π_i: Policy function, $\pi_i : \mathcal{O} \to \mathcal{A}$,
- θ_i: Learning parameters.

The market evolves in discrete time steps with price formation governed by:

$$p_t = f\left(\sum_{i=1}^{N} w_i \cdot \pi_i(o_t)\right) + \epsilon_t \tag{12}$$

where w_i denotes trading volume weights and ϵ_t is market noise.

9.0.2. FinCon Architecture

[7] introduces a multi-LLM architecture employing verbal reinforcement for reasoning refinement, formalized as:

Algorithm 1 AML Agent Workflow

```
Input: Transaction T
Alert ← TransactionMonitoring(T)
RiskScore ← NN_AML(Alert)
if RiskScore > θ then
    Investigation ← MultiAgentReview(Alert)
    Decision ← EnsembleVote(Investigation)
else
    Decision ← NoAction()
end if
Output: Decision
```

9.1. Specialized Trading Architectures

9.1.1. Multimodal Foundation Agent

[5] proposes a tool-augmented trading agent with multimodal feature fusion:

$$\hat{a}_t = \text{softmax}(W_\phi \cdot [f_{\text{text}}(x_t); f_{\text{chart}}(y_t); f_{\text{news}}(z_t)]) \tag{13}$$

where:

- f_{text}: Textual data encoder,
- f_{chart}: Technical analysis encoder,
- f_{news}: News sentiment encoder,
- W_ϕ: Learnable fusion weights.

9.1.2. FinRobot Platform

[34] introduces a modular, layered architecture:

$$\mathcal{F} = \mathcal{L}_{\text{data}} \oplus \mathcal{L}_{\text{LLM}} \oplus \mathcal{L}_{\text{agent}} \oplus \mathcal{L}_{\text{app}} \tag{14}$$

with each layer defined as:

- Data Layer $\mathcal{L}_{\text{data}}$: $\{\text{market}, \text{fundamental}, \text{alternative}\}$,

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 May 2025 doi:10.20944/preprints202505.0971.v1

- LLM Layer $\mathcal{L}_{\text{LLM}}$: $\{\text{general}, \text{financial-finetuned}\}$,
- Agent Layer $\mathcal{L}_{\text{agent}}$: $\{\text{single}, \text{multi}, \text{hybrid}\}$.

9.2. Risk Management Architectures

9.2.1. Agentic AI for Credit Risk

[40] proposes a hierarchical model for credit risk evaluation:

$$R(\mathbf{x}) = g\left(\sum_{j=1}^{k} \alpha_j h_j(\mathbf{x})\right) \tag{15}$$

where:

- h_j: Specialist risk sub-models (e.g., market, credit, operational),
- α_j: Attention weights derived from agent interactions,
- g: Final risk scoring function.

9.2.2. AML Agent Architecture

The WorkFusion system [54] implements an AML pipeline using agent collaboration:

Algorithm 2 AML Agent Workflow

Input: Transaction T
Alert $\leftarrow$ TransactionMonitoring(T)
RiskScore $\leftarrow$ NN_{AML}(Alert)
if RiskScore $> \theta$ **then**
    Investigation $\leftarrow$ MultiAgentReview(Alert)
    Decision $\leftarrow$ EnsembleVote(Investigation)
else
    Decision $\leftarrow$ NoAction()
end if
Output: Decision

9.3. Architectural Comparisons

Cloud computing platforms provide diverse Python libraries for developing AI agents, as summarized in Table 2.

Table 2. Architectural Features Comparison.

Paper	Type	Key Innovation	Math Foundation
[33]	Market Sim	Agent-based price formation	Game Theory
[7]	Multi-LLM	Verbal reinforcement loop	Ensemble Learning
[5]	Trading	Multimodal fusion	Attention Mechanisms
[40]	Risk	Hierarchical scoring	Neural Networks

10. Proposed Architectures: Mathematical and Algorithmic Foundations

Recent literature introduces a variety of architectures for agentic AI, each with unique mathematical and algorithmic principles.

10.1. Agent-Native and Modular Architectures

Agent-native foundation models are designed for multi-step planning, dynamic tool use, and memory integration [1–3]. These models enable agents to adaptively allocate computational resources, which can be expressed as:

$$y = f(x; \theta, \mathcal{A})$$

where x is the input, θ are model parameters, and $\mathcal{A}$ represents agentic actions or tools invoked during reasoning [1].

10.2. Meta-Agent and Hierarchical Planning

Meta-agent architectures introduce a supervisory agent that coordinates specialized sub-agents, optimizing for global objectives. This can be formalized as a hierarchical optimization problem:

$$\min_{\{\pi_i\}} \sum_{i=1}^{N} C_i(\pi_i) \quad \text{s.t.} \quad \bigcup_{i=1}^{N} \pi_i \in \mathcal{G}$$

where π_i is the plan for agent i, C_i is its cost, and $\mathcal{G}$ is the set of global goals.

10.3. Learning Agents and Reinforcement Learning

Learning agents adapt their behavior through feedback, often using reinforcement learning (RL) or RL from human feedback. The RL objective is:

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r_t \mid \pi\right]$$

where π is the policy, r_t the reward at time t, and γ the discount factor.

10.4. Automated Agent Design

Automated agent design leverages evolutionary algorithms to search the space of agent architectures. The process is:

$$\theta^* = \arg\max_{\theta} \mathcal{F}(\theta)$$

where θ encodes an agent architecture and $\mathcal{F}$ is a fitness function measuring performance.

10.5. Multi-Agent Coordination

Multi-agent systems distribute tasks and coordinate via protocols such as negotiation or centralized planning. The coordination can be modeled as:

$$\min_{\{\pi_k\}} \sum_{k=1}^{K} C_k(\pi_k) \quad \text{s.t.} \quad \text{Coord}(\{\pi_k\}) = \text{True}$$

where each agent's plan π_k must be compatible with others.

10.6. Algorithmic Example: Hierarchical Agent Planning

A recursive algorithm for hierarchical agent planning is described:

```
function HierarchicalPlan(goal)
    if Atomic(goal) then
        return Execute(goal)
    else
        subgoals ← Decompose(goal)
        for all subgoal in subgoals do
            result ← HierarchicalPlan(subgoal)
        end for
        return Aggregate(results)
    end if
end function
```

 doi:10.20944/preprints202505.0971.v1

10.7. Summary

The agentic AI field is evolving from monolithic LLMs to modular, hierarchical, and learning-enabled architectures, grounded in formal mathematical and algorithmic principles [1–3].

11. Implementation Challenges

11.1. Workforce Transformation

- Gartner predicts 80% of engineers need AI upskilling by 2027 [62]
- IBM emphasizes strategic AI upskilling [63]
- KPMG survey shows skills gap concerns [64]

11.2. Risk Alignment

Agentic systems require careful risk management:

- [38] examines alignment challenges
- [44] analyzes financial stability implications
- [61] provides executive playbook for adoption

11.3. Operational Considerations

- [8] outlines agent deployment strategies
- [46] tracks the rise of AI agents
- [65] notes capability expansion beyond productivity

12. Conclusion

AI agent frameworks are rapidly evolving, with significant implications for the financial industry. The combination of LLMs, multi-agent orchestration, and domain-specific integrations is enabling new levels of automation and intelligence. As the ecosystem matures, collaboration between academia, industry, and open-source communities will be key to realizing the full potential of agentic AI.

This paper has presented a novel multi-agent system architecture for advanced financial analysis. Our architecture leverages the power of LLMs and agentic AI to enable the development of intelligent financial agents that can collaborate to solve complex problems.

This paper presents a comprehensive exploration of multi-agent systems (MAS) within the context of financial analysis. We argue for a paradigm shift towards "orchestrated intelligence," where MAS, empowered by Large Language Models (LLMs) and sophisticated agentic AI frameworks, can revolutionize financial decision-making. We delve into the critical aspects of agent design, communication, and coordination, drawing upon recent advancements in AI agent frameworks [9,11,27] and the transformative potential of agentic AI in reshaping financial services [1,32,66]. Our proposed architecture addresses key challenges, including data integration, explainability, and risk management, and we present a detailed evaluation strategy to assess its efficacy in complex financial scenarios.

The survey reveals rapid advancement in AI agent frameworks and their financial applications. Key findings include:

1. Specialized frameworks (CrewAI, LangGraph) outperform general solutions for financial use cases
2. Productivity gains of 50-80% are achievable in data-intensive tasks
3. Risk management and trading show particularly strong benefits
4. Workforce transformation remains the largest adoption barrier

Future work should address standardization and safety in financial agent systems. As [67] notes, agentic AI represents both opportunity and disruption for the financial sector.

This comprehensive survey has examined the rapidly evolving landscape of AI agent frameworks and their transformative impact on financial services. Our analysis of 30+ recent publications (2024-2025) reveals three fundamental insights:

 doi:10.20944/preprints202505.0971.v1

First, modern agent frameworks like LangGraph, CrewAI, and AutoGen have matured to support mission-critical financial applications, demonstrating 50-80% efficiency gains in data-intensive tasks such as risk assessment [40] and trade execution [5]. The emergence of specialized architectures for financial markets [33] and risk management [54] underscores the domain-specific optimization required for production deployment.

Second, successful adoption requires addressing four key challenges: (1) workforce transformation through AI upskilling [62], (2) risk alignment in autonomous decision-making [38], (3) regulatory compliance in sensitive financial operations [44], and (4) integration with legacy systems [20]. Cloud-native Python libraries [26] and modular frameworks [34] are lowering these barriers.

Third, our theoretical analysis establishes formal foundations for agentic AI in finance, including:

- Market microfoundations via agent-based modeling [33]
- Multimodal fusion for trading systems [5]
- Hierarchical risk assessment frameworks [6]

Future work should prioritize: (1) standardization of agent communication protocols, (2) development of testing benchmarks for financial agent systems, and (3) hybrid architectures combining human expertise with agent autonomy [65]. As the field progresses, the principles outlined in this survey will help financial institutions navigate the transition from experimental deployments to production-scale agentic AI solutions [61].

12.1. Challenges and Future Directions

Future work will focus on addressing the identified challenges, including scalability, real-time performance, regulatory compliance, and integration with existing systems. The use of AI agents in areas like wealth management [68] is a promising avenue for future research.

Deploying agentic AI in finance presents challenges such as integration with legacy systems, ensuring compliance, and managing operational risks [20,23,69]. There is also a need for standardization and best practices to ensure reliability and trustworthiness [60]. Open-source initiatives and cloud-native platforms are accelerating innovation but require careful evaluation for production use [21,70,71].

References

1. Pounds, E. What Is Agentic AI?, 2024.
2. Jadhav, B. What is Agentic AI?, 2024.
3. Winston, A. What are AI agents and why do they matter?, 2024.
4. Schalkwyk, P.v. Part 3 AI at the Core: LLMs and Data Pipelines for Industrial Multi Agent Generative Systems, 2024.
5. Zhang, W.; Zhao, L.; Xia, H.; Sun, S.; Sun, J.; Qin, M.; Li, X.; Zhao, Y.; Zhao, Y.; Cai, X.; et al. A multimodal foundation agent for financial trading: Tool-augmented, diversified, and generalist. In Proceedings of the Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2024, pp. 4314–4325.
6. Han, X.; Wang, N.; Che, S.; Yang, H.; Zhang, K.; Xu, S.X. Enhancing Investment Analysis: Optimizing AI Agent Collaboration in Financial Research. In Proceedings of the Proceedings of the 5th ACM International Conference on AI in Finance, 2024, pp. 538–546.
7. Yu, Y.; Yao, Z.; Li, H.; Deng, Z.; Cao, Y.; Chen, Z.; Suchow, J.W.; Liu, R.; Cui, Z.; Xu, Z.; et al. Fincon: A synthesized llm multi-agent system with conceptual verbal reinforcement for enhanced financial decision making. *arXiv preprint arXiv:2407.06567* **2024**.
8. Yee, L.; Chui, M.; Roberts, R.; Xu, S. Why agents are the next frontier of generative AI. Technical report, McKinsey Digital Practice, 2024. tex.dateaccessed: 2025-01-24.
9. Arya, S. Top 7 Frameworks for Building AI Agents in 2025, 2024.
10. AI Agent Frameworks Compared: LangGraph vs CrewAI vs OpenAI Swarm.
11. Top 5 Frameworks for Building AI Agents in 2024 (Plus 1 Bonus), 2024.
12. Chen, S.H. Computationally intelligent agents in economics and finance, 2007. Issue: 5 Pages: 1153–1168 Publication Title: Information Sciences Volume: 177.

 doi:10.20944/preprints202505.0971.v1

13. Agents.
14. LangGraph.
15. CrewAI.
16. AI Agentic Design Patterns with AutoGen.
17. crickman. Semantic Kernel Agent Framework (Experimental), 2024.
18. Agentforce: Create Powerful AI Agents Salesforce US.
19. Build an Autonomous AI Assistant with Mosaic AI Agent Framework, 2024.
20. What are compound AI systems and AI agents?
21. What is Vertex AI Agent Builder?
22. AI Agents Amazon Bedrock Agents AWS.
23. wmwxwa. AI agents and solutions Azure Cosmos DB, 2024.
24. AI Agent Development IBM watsonx.ai.
25. Agents PydanticAI.
26. pydantic/pydantic-ai, 2025. original-date: 2024-06-21T15:55:04Z.
27. Aydın, K. Which AI Agent framework should i use? (CrewAI, Langgraph, Majestic one and pure code), 2024.
28. G, A. Best 5 Frameworks To Build Multi Agent AI Applications.
29. AI Agent Frameworks: Choosing the Right Foundation for Your Business IBM, 2025.
30. Top 5 Free AI Agent Frameworks.
31. These 2 AI Agent Frameworks Appear to Be Dominating Headlines—But Which One's Better? HackerNoon.
32. How Agentic AI will transform financial services, 2024.
33. Microfoundations, M.A.N.; Nakagawa, K.; Hirano, M. A Multi-agent Market Model Can Explain the Impact of AI Traders in Financial. In Proceedings of the PRIMA 2024: Principles and Practice of Multi-agent Systems: 25th International Conference, Kyoto, Japan, November 18-24, 2024, Proceedings. Springer Nature, 2024, Vol. 15395, p. 97.
34. Yang, H.; Zhang, B.; Wang, N.; Guo, C.; Zhang, X.; Lin, L.; Wang, J.; Zhou, T.; Guan, M.; Zhang, R.; et al. FinRobot: An Open Source AI Agent Platform for Financial Applications using Large Language Models. *arXiv preprint arXiv:2405.14767* **2024**.
35. Rogerson, K. Sphere for Employees – Agentic AI Copilot for Financial Services, 2024.
36. Jadhav, B. How Agentic AI is Redefining Employee Productivity?, 2024.
37. Cognizant Neuro AI.
38. Clatterbuck, H.; Castro, C.; Morán, A.M. Risk alignment in agentic AI systems. Technical report, Rethink Priorities, 2024. tex.dateaccessed: 2025-01-24.
39. Why 45% of financial firms are turning to GenAI for risk management.
40. See, M. AI and gen AI developments in credit risk management. Technical report, International Association of Credit Portfolio Managers, 2024. tex.dateaccessed: 2025-01-24.
41. Embracing generative AI in credit risk McKinsey.
42. sinclair schuller, E.Y. Wielding the double-edged sword of GenAI.
43. Agentic AI – the new frontier in GenAI.
44. Artificial intelligence and machine learning in financial services. Technical report, Financial Stability Board, 2024. tex.dateaccessed: 2025-01-24.
45. Bank, E.C. Artificial intelligence: a central bank's view **2024**.
46. The rise of AI agents. Technical report, Moody's Analytics, 2023. tex.dateaccessed: 2025-01-24.
47. AI and GenAI.
48. internationalbanker. Navigating the Generative AI Frontier: Balancing Risk and Workforce Transformation in Banking, 2024.
49. Introducing llama-agents: A Powerful Framework for Building Production Multi Agent AI Systems — LlamaIndex Build Knowledge Assistants over your Enterprise Data.
50. Deploy Generative AI with NVIDIA NIM NVIDIA.
51. IBM watsonx.
52. Leveraging Retrieval Augmented Generation (RAG) in Banking: A New Era of Finance Transformation.
53. camel-ai/camel, 2025. original-date: 2023-03-17T21:41:54Z.
54. AI Agents: Ready to Fight Financial Crime at Your Fingertips.
55. Capitec Bank employees save more than 1 hour per week with Microsoft 365 Copilot and Azure Open AI Microsoft Customer Stories.

 doi:10.20944/preprints202505.0971.v1

56. Woodie, A. AI Agent Claims 80% Reduction in Time to Complete Data Tasks, 2025.
57. JPMorgan Chase rolls out AI assistant powered by ChatGPT-maker OpenAI.
58. Zetaris introduces Agentic AI for the financial services sector, 2024.
59. AI Agents.
60. AI Agent Index – Documenting the technical and safety features of deployed agentic AI systems.
61. Agentic AI: The new frontier in generative AI an executive playbook. Technical report, PricewaterhouseCoopers, 2024. tex.dateaccessed: 2025-01-24.
62. Gartner Says Generative AI will Require 80% of Engineering Workforce to Upskill Through 2027.
63. AI Upskilling Strategy IBM, 2024.
64. GenAI 2024 Survey.
65. GenAI Doesn't Just Increase Productivity. It Expands Capabilities., 2024.
66. Singh, S. Agentic AI in Banking: Transforming Financial Services.
67. Getty, Joel Martin, S.D.D. GenAI isn't a threat to your job; agentic AI is, 2024.
68. AI in Banking: Benefits, Risks, What's Next.
69. generative-ai-for-beginners/17-ai-agents/README.md at main · microsoft/generative-ai-for-beginners.
70. Introducing smolagents: simple agents that write actions in code., 2025.
71. Google Launches Mariner, A New AI Agent Based On Updated Gemini 2.0.