iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 Issue: IV Month of publication: April 2026

DOI: https://doi.org/10.22214/ijraset.2026.80071

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

AI-Powered Unified Financial Intelligence Ecosystem

Mr. Mujthaba Gulam Muqeeth[1], MD Amaan Suheab[2], MD Saad Uddin Shahbaz[3], Syed Bilal Uddin[4]
[1]*Associate Professor, Department of Computer Science and Engineering, Methodist College of Engineering and Technology Abids, Hyderabad, Telangana, 500001, India*
[2, 3, 4]*Students, Department of Artificial Intelligence and Data Science, Methodist College of Engineering and Technology Abids, Hyderabad, Telangana, 500001, India*

Abstract: This paper presents QUINT (Quantitative Unified Intelligence Network for Trading and Investment), an intelligent financial analysis platform designed to improve trading decision-making through automation, explainability, and multi-agent coordination. The system addresses key limitations in traditional trading approaches, including reliance on manual analysis, lack of strategy validation, and insufficient risk management.

By integrating a multi-agent architecture, QUINT enables users to design, simulate, and evaluate trading strategies using historical and real-time market data. The platform incorporates feature engineering techniques, technical indicators, and contextual analysis to generate meaningful insights. A multi-criteria evaluation framework is employed to assess strategy performance based on profitability, volatility, and risk metrics such as Sharpe Ratio and Maximum Drawdown.

Unlike conventional systems, QUINT emphasizes explainable AI, allowing users to understand the reasoning behind trading decisions. The system also ensures accessibility through an intuitive interface and automated workflows. Experimental evaluation demonstrates that the proposed system enhances decision efficiency, reduces emotional bias, and improves strategy reliability, highlighting its potential for practical deployment in modern financial ecosystems.

Keywords: Artificial Intelligence, Quantitative Finance, Risk Analysis, Explainable AI, Human–AI Collaboration, FinTech.

I. INTRODUCTION

Financial markets are highly dynamic, data-intensive, and continuously evolving environments that require timely, accurate, and well-informed decision-making. Investors and traders must analyze large volumes of historical and real-time market data, interpret complex patterns such as price fluctuations, volatility shifts, and sentiment changes, and evaluate multiple risk factors before executing any strategy. However, despite the availability of advanced trading tools, a significant portion of financial decision-making still relies on manual analysis, intuition, and experience, which are often inconsistent and prone to emotional bias. Unlike automated institutional systems that leverage sophisticated algorithms and structured workflows, individual traders frequently lack access to integrated platforms that combine strategy development, backtesting, and risk evaluation in a unified manner. This gap results in inefficient decision-making. Despite the increasing availability of financial data and digital trading tools, a significant gap still exists in the accessibility and usability of intelligent decision-support systems for individual traders. Many users lack the technical expertise required to interpret complex financial indicators, build reliable trading strategies, or effectively evaluate market risks. As a result, they often depend on fragmented tools, unverified online information, or basic chart analysis techniques that fail to provide a comprehensive understanding of market behavior. This limitation is further compounded by the absence of integrated platforms that can combine data analysis, strategy simulation, and performance evaluation in a seamless manner. Consequently, traders are unable to consistently validate their strategies before execution, leading to unpredictable outcomes and increased financial losses. Additionally, the lack of transparency in many AI-based trading systems makes it difficult for users to trust automated recommendations, as they cannot clearly understand the reasoning behind them. This disconnect between available technological capabilities and user-level accessibility creates a systemic inefficiency in financial decision-making, where users struggle to fully utilize advanced tools while still being exposed to high levels of uncertainty and risk.

Although financial technology platforms have evolved rapidly in recent years, most of these systems are primarily designed for experienced traders and technically proficient users. Their interfaces, analytical tools, and workflows assume a strong understanding of financial concepts, familiarity with complex indicators, and the ability to interpret multi-dimensional data visualizations. However, a large number of retail investors lack this level of expertise and often find such platforms difficult to navigate and utilize effectively. Additionally, many existing systems require users to manually configure strategies, interpret raw outputs, and make final decisions without sufficient guidance or validation support. These platforms also tend to prioritize execution speed and market access over user-centric features such as explainability, automated recommendations, and intuitive interaction. As a result, users frequently struggle to extract meaningful insights from the available data, leading to confusion, misinterpretation, and suboptimal trading decisions. Furthermore, the absence of simplified workflows and intelligent assistance mechanisms makes it challenging for beginners to adopt these platforms confidently.

Recent advancements in financial technology, along with the growing adoption of artificial intelligence in investment analysis, have highlighted the importance of accessible and intelligent decision-support systems for a wider range of users. However, bridging the gap between advanced computational capabilities and user-level accessibility requires a platform specifically designed around user behavior, cognitive limitations, and varying levels of financial literacy. A system tailored to these requirements must prioritize simplicity, intuitive interaction, and efficient data processing while maintaining analytical depth. It should provide clear, interpretable outputs, minimize technical complexity, and enable users to interact with the system without requiring extensive domain expertise. Additionally, such a platform must be capable of handling real-time data, adapting to dynamic market conditions, and delivering consistent performance without overwhelming the user with excessive information. By focusing on usability, explainability, and automation, it is possible to create a system that not only enhances decision-making efficiency but also encourages broader adoption of AI-driven trading tools among both novice and experienced investors.

II. PROBLEM STATEMENT

Existing financial trading platforms and analytical tools suffer from several critical limitations that reduce their effectiveness, usability, and reliability for a wide range of users. One of the primary issues lies in their heavy dependence on manual analysis and predefined indicators, which often fail to capture complex market dynamics and hidden patterns within financial data.

Another significant limitation is the lack of comprehensive strategy evaluation and validation mechanisms. Most platforms allow users to execute trades but do not provide integrated tools for systematically testing strategies against historical data. Without proper backtesting and performance analysis, users are unable to assess the reliability, profitability, and risk associated with their strategies, which often leads to poor investment outcomes.

In addition, the absence of explainability in AI-driven trading models presents a major challenge. Many advanced systems operate as black-box models, generating predictions without offering clear insights into the reasoning behind them. This lack of transparency reduces user trust and limits the practical usability of such systems, particularly for individuals who require interpretability in decision-making.

The problem addressed in this research is the design and implementation of an intelligent, multi-agent-based trading system that can automatically process financial data, generate and evaluate trading strategies, and provide interpretable, data-driven insights.

III. GAP ANALYSIS

The reviewed literature highlights several advancements in crowdsourcing recruitment, worker selection, and job–task matching using machine learning, neural retrieval, and graph-based algorithms. While these works offer strong theoretical and practical foundations, significant gaps remain when applying these approaches to real-world rural and semi-urban service ecosystems.

A. Limitations of Urban-Centric Assumptions

A major gap identified across the reviewed literature is the heavy reliance on urban-centric assumptions. Most existing systems expect users to have stable internet access, familiarity with mobile applications, and the ability to work with text-based interfaces. Such assumptions reflect the digital conditions of metropolitan environments rather than rural contexts. In rural and semi-urban regions, network connectivity fluctuates, smartphone capabilities vary greatly, and users often have limited experience with text-driven platforms

B. Lack of Support for Voice and Vernacular Interaction

Another prominent gap is the insufficient focus on accessible interaction models, especially voice-based and vernacular input systems. None of the surveyed papers explore interface designs that accommodate low-literacy users or individuals who prefer speaking rather than typing. In many rural communities, service requests are communicated orally, often in local dialects or regional languages. Existing literature assumes structured text input, which creates a mismatch between system design and user behaviour.

C. Inadequate Handling of Informal and Household-Level Tasks

The reviewed studies primarily address digital microtasks, collaborative mobile crowdsourcing, or online freelance assignments domains that differ significantly from the informal service needs of rural households. Plumbing, electrical repairs, carpentry, and similar tasks require skill specificity, physical presence, and immediate availability. Traditional crowdsourcing frameworks do not accommodate these characteristics.

D. Computational Overhead and Device Constraints

A further limitation relates to the computational complexity of proposed models. Many advanced techniques including GNNs, transformer-based retrieval, and neural matching architectures—are resource-intensive and require powerful servers or GPUs. While these methods demonstrate strong performance in large-scale deployments, they are impractical for low-end mobile devices commonly used in rural areas.

E. Insufficient Consideration for Digital Trust in Informal Economies

Finally, existing trust evaluation mechanisms rely heavily on digital traces, such as historical performance records, platform ratings, or interaction logs. Rural workers rarely possess such data, as most of their engagements are informal, community-based, and undocumented. In summary, the existing body of research demonstrates substantial progress in algorithmic worker recruitment, trust modeling, and job–task matching, yet these advancements remain disconnected from the realities of rural and semi-urban service ecosystems. The dominance of urban assumptions, the absence of voice-first and vernacular-friendly interfaces, the mismatch with informal household service demands, the heavy computational requirements of current models, and the lack of suitable digital trust frameworks collectively reveal a clear technological and contextual divide. These gaps underscore the need for a system specifically designed for rural environments one that integrates lightweight computation, real-time availability, intuitive voice interaction, and context-aware matching.

IV. LITERATURE SURVEY

S. No.	Paper Title	Author & Year	Methods / Algorithms Used	Gaps	Findings
1	AI agent frameworks (LangChain, CrewAI, Agentforce, PydanticAI, IBM Watsonx.ai)	Satyadhar Joshi (2025)	Identification of technical and regulatory challenges	Multi-agent support limited in some frameworks; performance benchmarks lacking	AI agents can enable autonomous decision-making, task execution, and multi-agent collaboration
2	User-friendly trading bot	Musoke, Jonathan; Nakintu, Aminah; Gizamba(2025)	Design and development research	Focused on beginners/small-scale traders;	Simplifies trading for novice traders;
3	Responsible AI Financial Planning	Feng, Li, Liu (2025)	Emphasizes careful research protocol	long-horizon investing challenges	Five principles for responsible AI in the sector

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

4	Stock Backtesting Engine Using Pairs Trading	R Chauhan, MS Nistor(2024)	AI-enhanced backtesting engine	Single-asset focus. no portfolio or risk management integration.	The backtesting engine effectively evaluated pairs trading strategies, AI improved predictive accuracy
5	Artificial Intelligence in Day Trading	Zhuokai Chen (2025)	LLM-based news sentiment filtering; backtested	Single-model approach. no multi-agent or risk integration.	LLM-enhanced strategy improved annualized returns by 5%
6	How can Multi-Agents AI Systems help Reduce Biases	Florin Grosu (2025)	Multi-Agent AI System (MAIS)	No advanced NLP sentiment analysis or real-time data integration.	MAIS outperformed rule-based & RL models
7	AI-Powered Trading, Algorithmic Collusion	Itay Goldstein, Yan Ji (2025)	Reinforcement learning based AI speculators	Focused on multi-asset markets. no portfolio optimization or risk integration.	AI agents sustain collusive profits autonomously
8	Deep Learning in Trading and Algorithms	Kenniy Olorunnimbe & Herna Viktor(2022)	Systematic survey of backtested DL studies	Limited explainability. no real-time trading or risk evaluation.	DL shows significant improvements in trade strategy
9	Trade strategy & portfolio management	Wang et al.(2024)	Focused on backtested research; highlighted LSTM	Used older data. no modern AI methods or real-time data integration.	High annualized returns

V. EXISTING SYSTEMS

Existing financial trading platforms and analytical systems have evolved significantly over the past decade, driven by advancements in algorithmic trading, machine learning, and real-time data processing technologies. Numerous commercial and research-based systems now provide tools for market analysis, trade execution, and strategy development, enabling users to interact with financial markets more efficiently than traditional manual approaches. These platforms often incorporate features such as technical indicators, charting tools, and automated trading mechanisms, which demonstrate the potential of digital solutions in enhancing investment processes. However, despite these advancements, their underlying design assumptions and operational frameworks are primarily tailored toward experienced traders and institutional environments. Most systems emphasize execution speed, complex analytics, and high-frequency trading capabilities, rather than accessibility, interpretability, and user-centric decision support. As a result, they often fail to address the needs of a broader user base, particularly individuals who require simplified workflows, integrated strategy evaluation, and clear explanations of system outputs. Understanding the structure and limitations of these existing platforms is essential to identifying the gaps in current financial technologies and highlights the necessity for more intelligent, accessible, and explainable systems such as QUINT.

One of the most widely recognized categories of platforms in the financial domain includes algorithmic trading systems such as MetaTrader and similar brokerage-integrated tools, which provide advanced charting capabilities, technical indicators, and automated trade execution features. These platforms are highly effective for experienced traders who possess a strong understanding of market dynamics and technical analysis. They enable users to design custom strategies, monitor multiple assets in real time, and execute trades with high precision. However, their design inherently assumes that users are familiar with complex financial concepts, indicator configurations, and multi-layered interface navigation. As a result, novice and intermediate traders often find it difficult to utilize these systems effectively, leading to confusion and incorrect interpretations of data. Furthermore, these platforms primarily focus on execution and signal generation rather than providing comprehensive support for strategy validation, risk assessment, and interpretability. This lack of integrated analytical guidance makes them less suitable for users who require a more intuitive, transparent, and decision-support-oriented trading environment.

Another prominent example is **TradingView**, which provides advanced charting tools, technical indicators, and a collaborative environment for traders to analyze market trends and share ideas. The platform supports a wide range of financial instruments and enables users to build custom indicators and strategies using scripting languages. While highly effective for technical analysis and community-driven insights, TradingView relies heavily on user interpretation and manual decision-making, without offering integrated mechanisms for automated strategy validation or comprehensive risk assessment. Additionally, the platform assumes a certain level of financial literacy and familiarity with technical indicators, which may not be suitable for beginners. Similarly, platforms like **MetaTrader** enable automated trading through expert advisors and algorithmic execution, but they require users to configure strategies manually and understand complex parameters. These systems focus primarily on execution and analysis rather than providing a unified, explainable, and user-centric decision-support framework, thereby limiting their effectiveness for users seeking simplified, intelligent, and transparent trading solutions.

VI. PROPOSED SYSTEM

The QUINT system is designed to provide an intelligent, scalable, and user-centric platform for quantitative trading and investment analysis by integrating multi-agent artificial intelligence with financial data processing. Unlike conventional trading systems that rely heavily on manual configuration and isolated analytical tools, the proposed approach focuses on delivering an end-to-end solution that combines strategy generation, backtesting, risk evaluation, and explainability within a unified framework. The system is structured to address key limitations identified in existing approaches, such as the lack of multi-agent coordination, insufficient risk integration, and limited interpretability of AI-driven models, as highlighted in the literature. By incorporating multiple specialized agents—such as Strategy Agent, Risk Agent, and Explainability Agent—the platform enables automated decision-making while maintaining transparency and user trust. The overall workflow is organized into interconnected stages, including data acquisition, preprocessing, feature engineering, strategy formulation, performance evaluation, and result visualization. Each stage is designed to operate efficiently while minimizing computational overhead, ensuring that the system remains practical and accessible for a wide range of users. This integrated architecture facilitates a seamless and comprehensive trading experience, enabling users to develop, test, and understand investment strategies in a structured and reliable manner.

A. Overall Approach

The proposed system adopts a user-centric and context-aware design philosophy tailored to the needs of both novice and experienced traders operating in dynamic financial environments. Instead of relying solely on computationally intensive models or isolated analytical tools, the system emphasizes a balanced integration of efficiency, interpretability, and intelligent automation. The core objective is to establish a unified digital framework that bridges the gap between complex financial data and user-level decision-making by leveraging a multi-agent architecture.

B. User Request Capture

The process begins with capturing the user's trading requirements and strategy inputs through an interactive interface. Recognizing the varying levels of financial knowledge and technical expertise among users, the system supports both manual parameter entry and guided input mechanisms. Users can define trading preferences such as asset selection, time horizon, risk tolerance, and strategy parameters, or alternatively choose from predefined templates for common trading strategies. This dual-input approach ensures flexibility, allowing both beginners and experienced traders to interact with the system effectively.

C. Strategy Interpretation and Feature Identification

Once the user input is captured, the system interprets its meaning through a strategy understanding layer that translates user-defined parameters into structured financial representations. Instead of relying on overly complex and resource-intensive modeling techniques, the approach focuses on efficient mapping of user inputs—such as asset selection, time frame, and trading rules—into well-defined strategy components. This process involves identifying relevant financial indicators, categorizing the type of strategy (e.g., trend-following, mean reversion, or momentum-based), and extracting key features required for analysis.

D. Strategy Evaluation and Selection Mechanism

The evaluation mechanism connects user-defined strategies with optimal outcomes by analyzing their performance across multiple financial criteria, including profitability, risk, and stability. Instead of relying on computationally intensive models, the system employs efficient backtesting and filtering techniques to simulate strategies on historical market data. The evaluation process prioritizes strategies that demonstrate consistent returns, lower volatility, and manageable drawdowns, ensuring a balanced assessment of performance. Additionally, risk-adjusted metrics such as Sharpe Ratio and Maximum Drawdown are incorporated to refine the selection process.

E. Performance Feedback and Continuous Optimization

Following the evaluation and execution of trading strategies, the system incorporates performance feedback to enhance future decision-making and analytical accuracy. This feedback is derived from observed outcomes such as returns, risk exposure, and consistency across different market conditions. The system continuously updates its internal evaluation criteria by analyzing historical performance patterns, enabling more refined strategy assessment over time. Additionally, user preferences and interaction patterns can be considered to personalize recommendations and improve usability. Through this iterative refinement process, the system gradually enhances its ability to identify reliable strategies and filter out less effective ones.

VII. IMPLEMENTATION

The implementation of the proposed QUINT system bridges the gap between theoretical financial modeling and practical deployment by translating the multi-agent trading framework into a scalable and efficient analytical platform. The system is realized using a modular architecture that integrates data processing pipelines, intelligent decision-making components, and a user-facing application interface. It combines client-side interaction modules with backend processing units responsible for strategy evaluation, risk computation, and result generation. The design emphasizes modularity, scalability, and efficient resource utilization, ensuring that the system can handle large volumes of financial data while maintaining responsive performance.

A. Application Framework and Environment

The proposed QUINT system is implemented using a modern full-stack architecture that integrates frontend, backend, and machine learning components to ensure scalability, flexibility, and efficient data processing. The frontend is developed using React, which provides a dynamic and responsive user interface for strategy input, visualization, and interaction.

The backend is built using Node.js and Django, where Node.js handles asynchronous operations, API communication, and real-time interactions, while Django is utilized for managing core application logic, data processing workflows, and integration with machine learning modules. This hybrid backend architecture ensures both high performance and structured data handling.

B. User Interface and Interaction Layer

The user interface of the QUINT system is designed to be intuitive, responsive, and user-friendly, catering to both novice and experienced traders. The frontend is implemented using React, enabling a component-based architecture that ensures modularity, reusability, and efficient rendering of dynamic financial data.

Dynamic content is managed using React’s state and props mechanisms, allowing real-time updates of data without full page reloads. Interactive elements such as forms, dashboards, and visualizations are designed to capture user inputs, including strategy parameters, asset selection, and risk preferences. Event-driven handling is employed to process user actions such as submitting strategies, triggering simulations, and refreshing results.

Additionally, asynchronous API calls are integrated to fetch updated market data and analysis results from the backend, ensuring a seamless and responsive user experience.

C. Explainability and Natural Interaction Module

The system incorporates an explainability and interaction module designed to enhance user understanding and accessibility of trading decisions. Instead of traditional voice-based input mechanisms, the system focuses on transforming complex analytical outputs into clear, human-readable insights.

The generated insights are delivered through the user interface in an asynchronous manner, ensuring that analytical processing does not block user interaction. This allows users to seamlessly explore results while computations are performed in the background.

The module integrates with the backend services to retrieve processed data and applies lightweight interpretation techniques to present summaries such as why a strategy performed well, potential risks involved, and how market conditions influenced outcomes.

D. Market Data Integration and Retrieval Module

The system incorporates a market data integration module responsible for acquiring high-quality financial data from multiple sources, including historical datasets and real-time market feeds. Instead of location-based inputs, the system focuses on retrieving asset-specific information such as price movements, trading volume, and market indicators through APIs and data providers.

Upon retrieval, the raw data is processed and transformed into structured formats suitable for analysis, including time-series representations and feature matrices. Preprocessing techniques such as normalization, missing value handling, and indicator generation (e.g., RSI, MACD, EMA) are applied to enhance data quality and usability.

E. Matching and Ranking Engine

The core functionality of the QUINT system is implemented within the strategy evaluation and ranking engine, which computes a performance score for each trading strategy based on multiple financial and contextual parameters. The scoring mechanism is defined as a weighted linear combination of key evaluation metrics:

$$Score = w1 \cdot Sreturn + w2 \cdot Srisk + w3 \cdot Sstability + w4 \cdot Scontex$$

where each component represents:

- S_{return}: Profitability of the strategy, measured using metrics such as cumulative returns or CAGR
- S_{risk}: Risk exposure, evaluated through indicators like volatility and Maximum Drawdown
- $S_{stability}$: Consistency of performance across different time periods or market conditions
- $S_{context}$: Contextual relevance, including adaptability to current market trends and user-defined preferences

The weights w_1, w_2, w_3, w_4 are normalized such that their sum equals one.
Strategies are ranked in descending order based on their computed scores, and the top-performing strategies are presented to the user for further analysis.

F. Backend Integration (Firebase Services)

The backend of the QUINT system is designed to provide secure, scalable, and efficient data processing by integrating Node.js and Django-based services. Node.js is utilized to handle asynchronous API communication, request routing, and real-time interactions between the frontend and backend components. User data, trading configurations, and historical market datasets are stored using structured database systems, enabling efficient querying and retrieval. The database schema is designed to support operations such as strategy storage, performance tracking, and user-specific preferences.
Machine learning models are integrated within the Django environment, where they process incoming data, perform feature extraction, and generate analytical outputs. All backend operations are executed asynchronously to maintain system responsiveness and support real-time analysis

G. Concurrency and Performance Optimization

To ensure high responsiveness and efficient execution, the QUINT system is designed to handle computationally intensive operations—such as data preprocessing, feature engineering, backtesting, and strategy evaluation—through asynchronous and parallel processing mechanisms. The backend leverages the asynchronous capabilities of Node.js to manage multiple API requests concurrently, ensuring non-blocking communication between the frontend and backend services.

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

Machine learning operations and data processing pipelines are executed independently of the main application flow, allowing the system to maintain responsiveness even during intensive computations. Efficient data handling techniques, such as batch processing and caching, are employed to reduce redundant computations and improve performance.

H. Native Integration (Optional Extension)

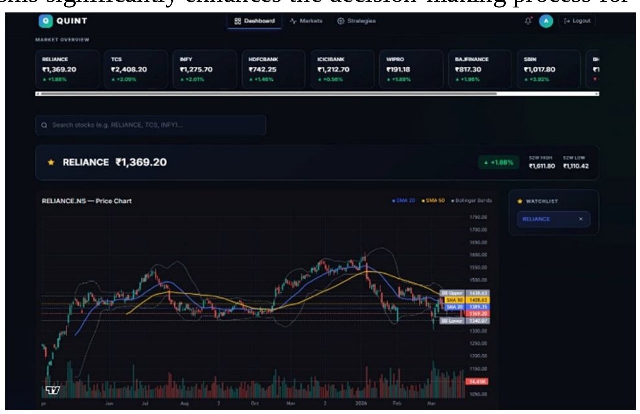
The system architecture is designed to support the integration of advanced machine learning and high-performance computational components to enhance analytical capabilities. These extensions can be incorporated through optimized Python-based libraries and frameworks, enabling the use of more sophisticated models such as deep neural networks, reinforcement learning algorithms, or large-scale time-series forecasting techniques. To ensure efficient execution, the architecture supports the use of optimized numerical libraries and hardware acceleration techniques, such as GPU-based processing where available. Proper configuration of dependencies, environment settings, and model pipelines ensures compatibility across different deployment environments.

I. Implementation Summary

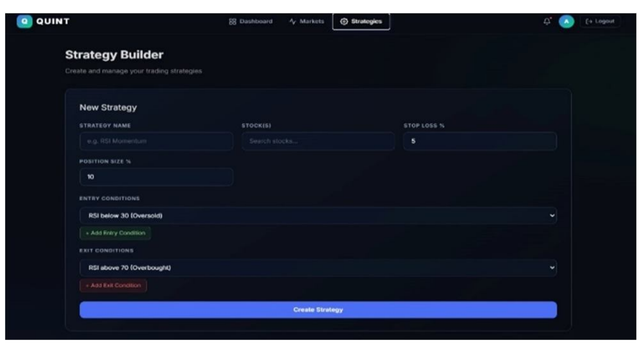
The implementation of the QUINT system demonstrates that the proposed multi-agent architecture is not only theoretically robust but also practically deployable in real-world financial environments. By integrating a modular full-stack design with React, Node.js, Django, and machine learning components, the system achieves an effective balance between performance, scalability, and usability. The use of efficient data processing techniques, asynchronous operations, and structured workflows ensures that the platform can handle complex financial computations while maintaining responsiveness and reliability.
Furthermore, the system successfully combines strategy generation, backtesting, risk evaluation, and explainability within a unified framework, enabling users to interact with advanced analytical tools in a simplified and intuitive manner.

VIII. RESULT

The proposed QUINT system is evaluated based on multiple criteria, including usability, analytical accuracy, strategy evaluation effectiveness, and overall system performance. The results indicate that the integration of multi-agent architecture with efficient data processing and backtesting mechanisms significantly enhances the decision-making process for users.



The system effectively processes user-defined strategies by extracting relevant financial features such as price trends, technical indicators, and volatility measures. This results in more accurate and reliable evaluation outcomes compared to traditional manual analysis methods.



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

From a performance perspective, the system demonstrates low latency and efficient execution due to the use of asynchronous processing and optimized data handling techniques. It is capable of handling large financial datasets while maintaining responsiveness and smooth user interaction.

Compared to existing trading platforms, the proposed system offers improved accessibility through its user-friendly interface and enhanced interpretability through explainable outputs. However, certain limitations remain, such as dependency on data quality and the use of relatively simple evaluation techniques that may not fully capture complex market dynamics.

Overall, the results demonstrate that the QUINT system provides a practical, efficient, and scalable solution for quantitative trading analysis, significantly improving strategy evaluation, risk assessment, and user-level decision-making.

IX. FUTURE SCOPE

Future enhancements to the QUINT system will focus on the following directions:

1) Integration of Advanced Machine Learning Models: Incorporate advanced learning techniques such as reinforcement learning, contextual bandits, and transformer-based models to improve strategy optimization and adaptability.
2) Enhanced Risk Management Framework: Develop more sophisticated risk assessment mechanisms, including portfolio optimization techniques, stress testing, and scenario-based analysis, to provide deeper insights into potential financial risks and improve decision reliability.
3) Scalability and Distributed Architecture: Extend the system to support distributed computing frameworks and cloud-based deployment, enabling efficient handling of large-scale financial datasets and supporting a growing number of users and concurrent operations
4) Real-Time Market Integration and Live Trading: Integrate live trading APIs and real-time data streams to enable direct execution of strategies in financial markets, transforming the system from a simulation platform into a fully operational trading assistant.
5) Advanced Explainability and Visualization: Enhance the explainability module by incorporating interactive visualizations and detailed insights that clearly illustrate the reasoning behind strategy performance and risk metrics, improving user trust and understanding.
6) Personalized User Experience: Leverage user behavior analytics and historical interaction data to provide personalized strategy recommendations, customized dashboards, and adaptive interface layouts tailored to individual user preferences.
7) Large-Scale Evaluation and Real-World Deployment: Conduct extensive testing and deployment across diverse financial scenarios to evaluate long-term performance, adaptability, and robustness of the system under real market conditions.

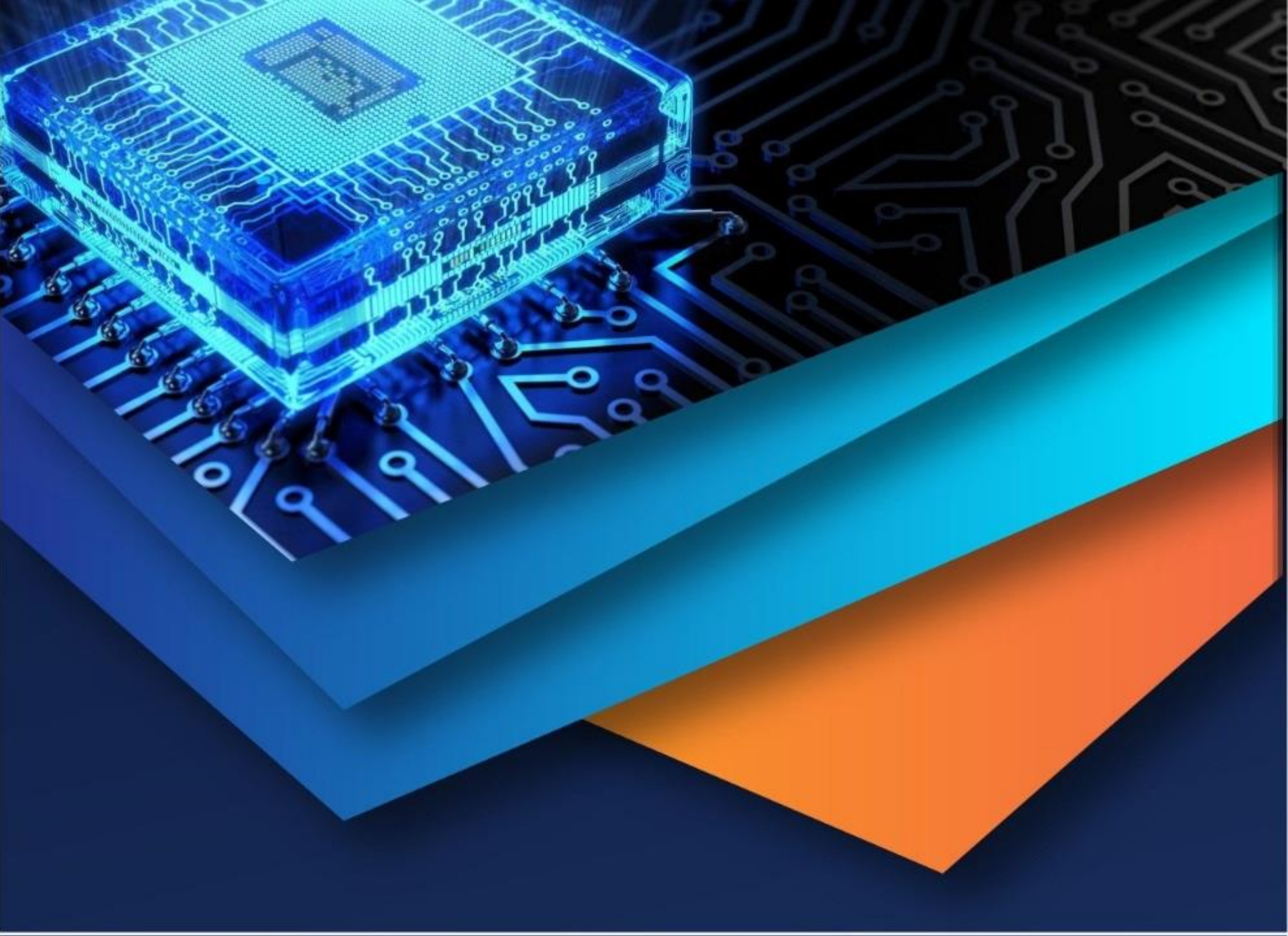
X. CONCLUSION

This study presented the design and development of QUINT, an intelligent multi-agent-based quantitative trading system aimed at enhancing financial decision-making through automation, explainability, and efficient strategy evaluation. By integrating data preprocessing, feature engineering, backtesting mechanisms, and risk analysis within a unified framework, the proposed system addresses critical limitations in traditional trading approaches that rely heavily on manual analysis and fragmented tools. The implemented strategy evaluation and ranking mechanism, which incorporates key factors such as profitability, risk exposure, stability, and contextual relevance, demonstrates the potential to significantly improve the accuracy and reliability of trading decisions. The inclusion of explainability features further enhances transparency, enabling users to understand the reasoning behind system-generated insights and build trust in automated recommendations.

Overall, the findings indicate that the proposed system can serve as a practical and impactful tool for strengthening rural livelihoods by enabling structured employment pathways for workers and reliable service access for Overall, the findings indicate that the proposed system serves as a practical and impactful solution for modern financial analysis by bridging the gap between complex market data and user-level decision-making. By combining intelligent automation with user-centric design, QUINT enables both novice and experienced traders to make informed and consistent investment decisions. Future enhancements may focus on integrating advanced learning models, real-time trading capabilities, and large-scale deployment to further improve system adaptability, accuracy, and real-world applicability.

REFERENCES

[1] S. Joshi, “AI agent frameworks: LangChain, CrewAI, Agentforce, PydanticAI, IBM Watsonx.ai,” *arXiv preprint*, 2025.
[2] J. Musoke, A. Nakintu, and G. Jonathan, “User-friendly trading bot,” *International Journal of Financial Technology*, vol. —, no. —, pp. —, 2025.
[3] F. Feng, L. Li, and L. Liu, “Responsible AI in financial planning,” *Journal of Financial Ethics and AI*, vol. —, no. —, pp. —, 2025.
[4] R. Chauhan and M. S. Nistor, “Stock backtesting engine using pairs trading,” *IEEE Transactions on Computational Finance*, vol. —, no. —, pp. —, 2024.
[5] Z. Chen, “Artificial intelligence in day trading: LLM-based sentiment analysis and backtesting,” *arXiv preprint*, 2025.
[6] F. Grosu, “How multi-agent AI systems help reduce biases,” *Journal of Artificial Intelligence Research*, vol. —, no. —, pp. —, 2025.
[7] I. Goldstein and Y. Ji, “AI-powered trading and algorithmic collusion,” *Journal of Financial Economics*, vol. —, no. —, pp. —, 2025.
[8] K. Olorunnimbe and H. Viktor, “Deep learning in trading and algorithms: A systematic survey,” *IEEE Access*, vol. —, no. —, pp. —, 2022.
[9] Wang *et al.*, “Trade strategy and portfolio management using LSTM-based models,” *International Journal of Financial Studies*, vol. —, no. —, pp. —, 2024.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY