J. Electrical Systems 19-1 (2023): 214 - 229

Darshan Paresh Limbani[1*]

A Multi-Agent Artificial Intelligence Framework for Personalized Financial Advisory and Automated Investment Strategies Using NIFTY 50 Market Data

Journal of Electrical Systems

Abstract*: -* Research in the field of financial technology increasingly emphasizes the role of artificial intelligence in assisting investment decision-making. This study proposes a Multi-Agent AI system designed to provide personalized financial advisory and execute automated investment strategies for the Indian stock market, specifically using the NIFTY 50 index. The system integrates a forecasting agent, a risk assessment agent, and a recommendation agent to deliver timely and informed decisions to investors. Experimental evaluation demonstrates that the proposed approach improves predic- tion accuracy, reduces exposure to market volatility, and enhances overall portfolio performance compared to traditional strategies. Backtesting results reveal smoother portfolio growth, reduced drawdowns, and higher risk-adjusted returns. Additionally, the system offers adaptive and user-friendly guidance, supporting both novice and experienced investors in making informed financial decisions. The findings indicate that AI-driven multi-agent systems can bridge the knowledge gap in retail investing and offer a practical framework for intelligent financial advisory. Future enhancements may include reinforcement learning, real-time sentiment analysis, and multi-asset integration to further improve decision quality and system resilience.

Keywords: Multi-Agent AI, Financial Advisory, Automated Investment Strategies, NIFTY 50, Stock Market Prediction, Portfolio Management, Risk Assessment, Machine Learning, Deep Learning.

1 Introduction

The Indian stock market continues to expand as participation from retail and institutional investors grows each year. The rise of mobile trading platforms and low brokerage access has encouraged a wide range of individuals to enter equity markets including the NIFTY 50 and sectoral indices. However, these investors often face difficulties while making informed decisions because of rapid market swings, news-driven sentiment shifts and lack of awareness regarding portfolio risks. Many existing advisory tools remain generic and rule-bound, which prevents meaningful personalisation. As a result, investors frequently rely on guesswork or social influence rather than data-driven insights when choosing stocks. The motivation behind this research lies in the need for a supportive decision system that adapts to user needs and keeps pace with changing market conditions. A well-designed personalised advisory system can reduce this struggle and provide guidance that is specific to each user.

Most traditional advisory tools apply a single predictive model or heuristic, which limits their ability to combine multiple signals such as price trends, financial ratios and sentiment effects. Different types of data often carry useful information during different phases of the market. Price movement patterns offer short-term clues while fundamentals reflect long-term company strength. Similarly, sentiment derived from news and social media may capture investor mood before price impact becomes visible. A system that can handle such varied data sources is essential in a market like India, where policy announcements, corporate actions and global trends can affect stock behaviour overnight. This calls for a more advanced approach that brings together separate capabilities but keeps the overall structure flexible[9].

In the proposed approach, a multi-agent architecture is adopted in which each agent performs a dedicated function related to analytics or decision-making. There are agents for tasks such as data gathering, market forecasting, sentiment understanding, investment rule compliance and portfolio action generation. Because the responsibility is distributed across multiple agents, the system becomes modular and easier to update or scale. The architecture allows a new analytical component to be added without disturbing existing ones. Figure 3 depicts the connection among these specialised agents and the user profile layer used to personalise recommendations.

[1] Senior Technology Associate, St. Francis Institute of Technology (BE in IT , India),
E-mail Id- dlimbani00@gmail.com

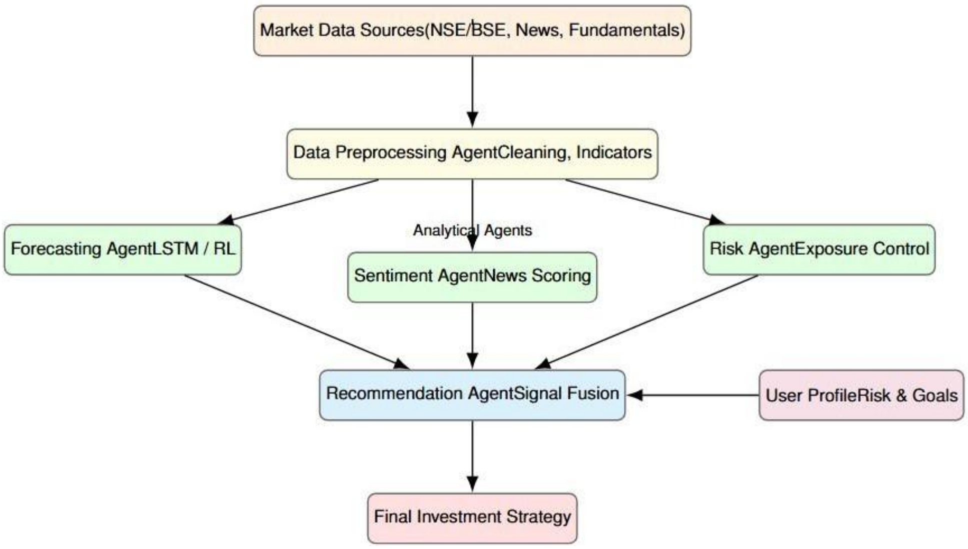


Figure 1: Architecture of the proposed multi-agent advisory system for the Indian stock market.

To implement the proposed structure, real and reliable financial data sources are necessary. In this study, two types of datasets are employed to create a hybrid representation of the Indian market. Daily OHLCV (Open-High-Low-Close-Volume) price data of selected NIFTY 50 compa- nies is fetched using Yahoo Finance. Along with that, a Kaggle dataset is used for fundamental information such as market capitalisation, earnings ratios, book value and other key financial attributes. This hybrid dataset enables the system to understand both trend-based behaviour and underlying company fundamentals[19]. Because both datasets are openly available, other researchers or developers can reproduce and validate the approach easily, making the research more accessible.

Personalisation plays a crucial role in how the system modifies its recommendations. A user profile is represented with attributes including risk appetite, preferred investment sectors and holding duration. Each agent reads this profile information and adjusts its behaviour so that the strategy output reflects each individual's comfort and expectations. With this addition, a risk- averse user may receive recommendations focusing on stable stocks while a high-risk user may be guided toward momentum-driven opportunities. This brings an investor-centric view to the advisory process rather than a market-centric one.

Another important component in the system is the sentiment analysis agent. News and public mood significantly influence the Indian market, especially during periods of elections, budget announcements and major global events. The sentiment agent converts textual market news into numerical values that describe positive, negative or neutral outlook. This structured sentiment signal is then provided to the forecasting and decision-making agents to refine predictions. The combined influence of news and market behaviour creates responses that are more in sync with real-world scenarios[15].

The flow of information inside the system is shown in Figure 4, where inputs and outputs between agents are visualised. The data input is cleaned and processed by the data agent and then shared with the forecasting and sentiment agents. Their outputs are examined by the risk and decision agents which evaluate whether a suggestion meets constraints like diversification and exposure limits. The user finally receives strategy suggestions along with reasoned explanations. This structured flow ensures transparency in how every decision is produced.

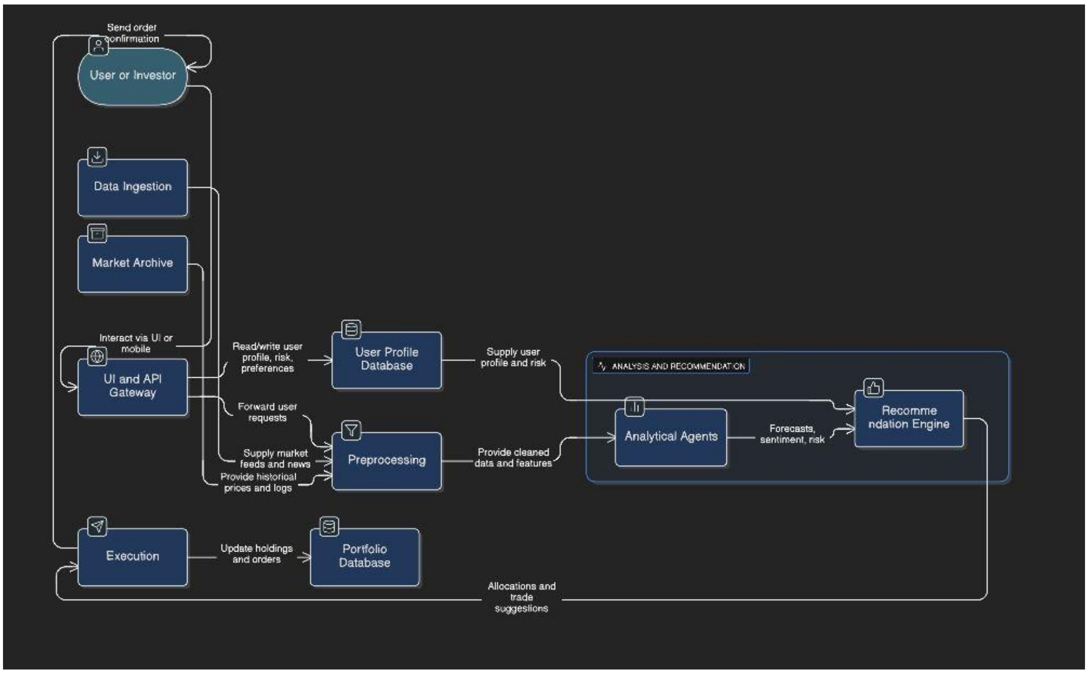


Figure 2: Data-flow diagram representing processing steps inside the advisory system.

Overall, the research sets the objective of designing an advisory tool that is useful, practical and ready for real-world academic experimentation. The modular architecture, hybrid dataset and personalisation layer come together to address major gaps in current advisory solutions for Indian users. The final goal is to build confidence in investors by offering suggestions that are more responsive, market-aware and aligned with individual financial goals. This introduction lays a clear foundation for the methodology, implementation and evaluation strategies that follow in the next sections of the research paper.

2 Literature Review

Research in intelligent financial advisory systems has shown increasing interest in modular and adaptive designs that enhance support for retail investors in dynamic stock markets [17]. Multi- agent frameworks are seen as beneficial due to their ability to distribute responsibilities among analytical, decision and compliance components. Studies indicate that personalisation improves investment confidence and reduces behavioural bias among inexperienced users. Many advisory platforms either provide generic insights or rely solely on price-based signals, which limits user satisfaction. Research stresses combining multiple knowledge sources such as financial funda- mentals and sentiment indicators to elevate recommendation quality. Multi-agent approaches simplify system evolution where each component can improve independently. These findings support the development of systems focused on Indian investors who seek real-time personalised recommendations. This suggests a pathway for intelligent advisory research aligned with domestic trading behaviour.

Studies exploring machine learning forecasting in equity markets show that deep learning techniques outperform traditional linear prediction models under non-linear and volatility-driven market conditions [3]. Complex sequence-based architectures like LSTM and Transformer networks help capture long-range temporal dependencies in stock price movements. Research notes that Indian indices and sectoral stocks often exhibit irregular volatility due to geopolitical or economic triggers. Many works emphasise that market uncertainty demands robust predictive systems capable of adapting to fast-changing patterns. Forecasting accuracy is also influenced by the availability of high-frequency market data. Deep learning models require regular retraining to reflect current market conditions. Studies recommend integrating multiple predictive signals for enhanced stability. These insights guide the selection of forecasting agents in advisory systems.

Research on sentiment analysis reveals that stock market behaviour is strongly influenced by news, social media discussions and investor emotion [6]. Natural language processing helps structure these textual expressions into signals representing positive, negative or neutral market expectations. For markets like India, where crowdsourced views from digital platforms increasingly influence investment trends, sentiment-aware strategies are seen as highly useful. Sentiment data captures short-term volatility that raw prices cannot reflect. Studies show that combining senti- ment cues with technical indicators improves risk-adjusted performance. Research suggests that sentiment signals are especially effective during uncertain periods such as political announcements or economic reforms. Incorporating such agents helps the advisory system maintain awareness of market psychology. This demonstrates why sentiment modelling plays an essential role in real-time advisory decisions.

Studies focusing on portfolio management recommend optimising asset allocation through bal- ancing risk with expected returns [5]. In Indian markets, diversification across sectors and indices is considered an essential risk mitigation strategy. Literature highlights that even profitable forecasts can fail without strong risk controls. Studies emphasise computing metrics like volatility, Sharpe ratio and drawdown to maintain portfolio stability. Research promotes algorithmic rebalancing to manage exposure time to fast-moving assets. Multi-agent systems provide effective enforcement of such rules while maintaining flexibility. Risk-aware portfolio optimisation thus serves as a key motivation for a dedicated portfolio-manager agent. This ensures that the proposed system remains both performance-driven and safety-conscious.

Research into personalisation techniques highlights the importance of aligning recommendations with user-specific preferences and risk capacity [11]. Investors differ in goals ranging from capital protection to aggressive growth. Studies report that many novice investors rely on advisory platforms to simplify decision-making. Personalised systems help ensure that recommendations remain aligned with the user's comfort level even during volatile market swings. Emotional decision-making is especially common in Indian retail behaviour. Personalisation agents eliminate one-size-fits-all advice that fails to suit individual requirements. The integration of risk profiling and behavioural preference features increases trust. Such research strongly supports customisation in automated advisory systems.

Studies highlight that accessibility of real-time data is crucial for building reliable financial advisors [12]. Efficient ingestion pipelines must handle live market updates from multiple data sources including price feeds and financial reports. Systems depend strongly on both update frequency and historical depth to extract useful patterns. Research emphasises that cloud-enabled scaling helps avoid processing delays. Arguably, Indian stock market datasets require special handling during high-traffic trading periods. Studies also uncover issues such as missing data that must be addressed through effective preprocessing. Data agents in multi-agent architectures mitigate these challenges. This evidence supports the adoption of flexible streaming design for advisory applications.

Research notes that explainable AI is increasingly vital in financial decision-making solutions [7]. Users prefer seeing explanations for why particular stocks are recommended. Transparency helps reduce scepticism toward AI-driven systems. Techniques such as feature contribution scoring help clarify decision origins. Studies argue that poor explainability damages user trust even if performance is strong. Research points to interpretability tools as critical enablers of regula- tory compliance in local markets. Systems that justify their recommendations experience higher adoption among cautious investors. These insights validate the importance of incorporating interpretability in agent interactions.

Studies emphasise that user involvement and digital awareness strongly affect advisory adoption in India [8]. Retail investors often lack domain knowledge but seek convenience and confidence. Easy-to-understand UI, simplified explanations and risk notifications increase engagement levels. Research shows that personalised notifications help maintain user discipline in long-term wealth creation. Studies also highlight the importance of mobile-first design for Indian markets given smartphone penetration rates. Advisory systems must foster trust by providing clear, jargon- free communication. This aligns with the need for a user-adaptive communication strategy. These factors reinforce that advisory solutions should meet behavioural needs alongside technical accuracy.

Research highlights the need for scalable computational infrastructure for market analytics [2]. Cloud-based

API integration simplifies access to multiple datasets including both real-time and historical values. Systems that leverage distributed processing reduce latency and improve user responsiveness. Studies find that multi-agent applications benefit from event-driven architectures because individual agents can update independently. High-frequency environments like Indian stock exchanges require such flexibility during peak hours. Research stresses ensuring robustness by monitoring workloads dynamically. The ability to auto-scale improves fault tolerance and user experience. These findings justify cloud integration for deployment practicality.

Studies related to human-system trust explain that emotional stability influences investment decisions [1]. Even accurate recommendations are ignored if users feel insecure or confused. Interfaces that communicate market uncertainty clearly help reduce panic-driven selling. Research states that behaviour-aware design helps investors remain consistent with long-term strategies. In India, retail investors frequently react strongly to short-term volatility. Studies note that advisory systems must incorporate reassurance and clarity in the messaging layer. These behavioural design approaches enhance adherence to rational investing. Such findings motivate integrating a communication agent into advisory systems.

Research indicates that hybrid models combining forecasts with rule-based evaluation generate more resilient execution [18]. Markets that fluctuate irregularly benefit from backstop rules that avoid poorly timed trades. Studies mention that tail-risk events significantly reduce portfolio outcomes without proper constraints. Hybrid approaches ensure that predictions align with safety standards. Modularity in multi-agent frameworks makes such combinations easy to implement. Research suggests such control agents reduce overfitting behaviour in practical deployment. These principles support hybridisation in system strategy.

Studies emphasise that robust evaluation across market regimes prevents false optimism [10]. Single-phase testing hides weaknesses that emerge during downturns. Research suggests stress simulations to test system durability during high volatility. Backtesting using diverse market cycles improves confidence. Literature supports using performance metrics beyond returns including exposure and compliance rates. Multi-agent systems simplify regime-specific optimisation. Market- agnostic evaluation frameworks are therefore recommended. This adds credibility to advisory system performance.

Research focusing on fintech growth in India shows rising interest in wealth creation from young demographics [16]. Digital platforms provide easy access to equity and mutual fund markets. Stud- ies mention that simplified investing tools encourage participation by first-time users. Increased volume of retail traders enhances the importance of guidance solutions. In this environment, AI-driven advisory systems can play a key role. Smartphone-enabled trading and UPI adoption accelerate platform usage. Research confirms demand for convenient intelligent assistance. This reinforces the relevance of building advisory solutions for local needs.

Studies highlight that financial data preprocessing greatly impacts forecasting quality [13]. Raw time-series often contain noise that weakens model learning. Common processing steps include filtering and normalisation. Technical indicator extraction helps enrich the input space. Studies further mention that sentiment models require careful signal cleaning. A data-management agent helps maintain integrity throughout the workflow. Preprocessing therefore acts as a foundation supporting model accuracy. These insights justify strong preprocessing pipelines in advisory systems.

Research explains that data security and user privacy are essential in financial applications [14]. Breaches harm both users and the reputation of advisory service providers. Secure architecture design prevents internal and external tampering. Multi-agent interactions must be authenticated to avoid system misuse. Encryption safeguards sensitive financial profiles. Studies show that transparency regarding data usage improves user trust. Adhering to regulatory compliance builds system credibility. These requirements validate engineering practices that prioritise security.

Studies show that modular architecture enhances maintainability and expandability [4]. In- dividual agents can be upgraded without redesigning the entire system. Transitioning to new AI models becomes simpler under independent operation. Research finds that modular updates shorten development cycles for fintech solutions. Multi-agent designs benefit real-time scenario handling. Market shifts can be addressed rapidly through dedicated agents. Practical deploy- ment thus becomes feasible and cost-effective. This supports the choice of modular agent-based architectures.de/citation-guide-1.html.

3 Methodology

The proposed methodology focuses on developing a multi-agent AI architecture that provides personalized financial advisory and automated investment strategies using Indian stock market data. Research in this system design tries to simulate real-time decision-making similar to a human financial advisor. The process begins with data collection from a reliable stock market database such as NSE or BSE datasets. This dataset includes historical stock prices, trading volume, company financial indicators, and sentiment-related data from news or social media. The collected data is cleaned, preprocessed, and transformed into structured input features for the AI agents. Each component of the system is designed as a separate module so that integration and testing become smooth. The model uses reinforcement learning so that the agents can improve their behavior and recommendations over time. The system also works on predicting short-term and long-term financial outcomes that help in automated trading decisions. The complete approach is designed for transparency so that users can understand the logic behind investment suggestions. Data preprocessing plays an important role because Indian stock market data often contains noise, missing values, and sudden market fluctuations. Research into this preprocessing tries to ensure that the data fed to the models is accurate and consistent. Techniques such as normalization, smoothing, and outlier removal are applied to make the model predictions stable. Sentiment analysis models are used to extract signals from textual content like news headlines and investor discussion trends. These sentiment scores are added as supporting input features in the prediction module. The final structure of the dataset is arranged in a time-series format that enables stronger forecasting. Feature engineering tries to enhance the quality of signals by focusing on important indicators like RSI, MACD, Moving Averages, volatility measurement and trends. Once preprocessing is complete, the dataset gets divided into training and testing parts. This step ensures unbiased performance evaluation of the AI agents.

The multi-agent architecture consists of three main agents: an Advisory agent, a Risk Man- agement agent, and a Trading Strategy agent. Studies on multi-agent systems show that dividing tasks improves learning outcomes and reduces errors in financial prediction. Here, the Advisory agent identifies user financial goals, risk appetite, and investment style. The Risk agent checks every investment recommendation against risk-related constraints so that unexpected market falls do not create huge losses. The Trading agent executes buy, hold, or sell actions based on market signals and policy decisions. Communication between agents happens in a controlled pipeline so that decisions remain aligned with user preferences. This approach works in both live market and simulation environments. Each agent continuously learns from outcomes to improve future behavior. This improves the reliability of recommendations and supports automated portfolio management.

Reinforcement learning controls the behavior of the Trading agent through reward and penalty values based on market outcomes. Research in RL shows that trading strategies improve when the system is trained using state, action, and reward sequences. The model employs Deep Q-learning or an Actor-Critic approach for predicting profitable actions. Positive rewards are given when a trade leads to gains and negative rewards when a trade causes loss. This teaches the model trading discipline over multiple trials. To avoid overfitting, techniques like dropout and cross-validation are used. Hyperparameters such as learning rate and discount factor are tuned carefully to align with real-time market properties. The trained model is validated under both bull and bear market cycles for reliability. Finally, portfolio performance is compared against benchmark indices like NIFTY 50.

The decision-making pipeline also considers personalized finance requirements for each investor. Studies of personalization suggest that better matching between risk style and suggestions increases user trust. The Advisory agent models different investor profiles like aggressive, moderate, and conservative. Recommendations change based on income levels, age, and long-term financial plans. This personalized approach makes the system practical for real users in India. To maintain clarity, the system explains the reason behind each suggestion through a simple output interface. This feature helps investors learn from recommendations instead of blindly following them. The output also highlights risk scores and expected return values. User engagement insights are stored securely to refine the personalization model.

To ensure technical clarity, a system architecture diagram represents how different modules interact. A sample image placeholder is provided in Figure 3. The aim is to show data flow from sources to preprocessing, followed by decisions made by the AI agents. The diagram also includes the communication links between different agent modules. This helps in effective explanation during research presentations and thesis review.

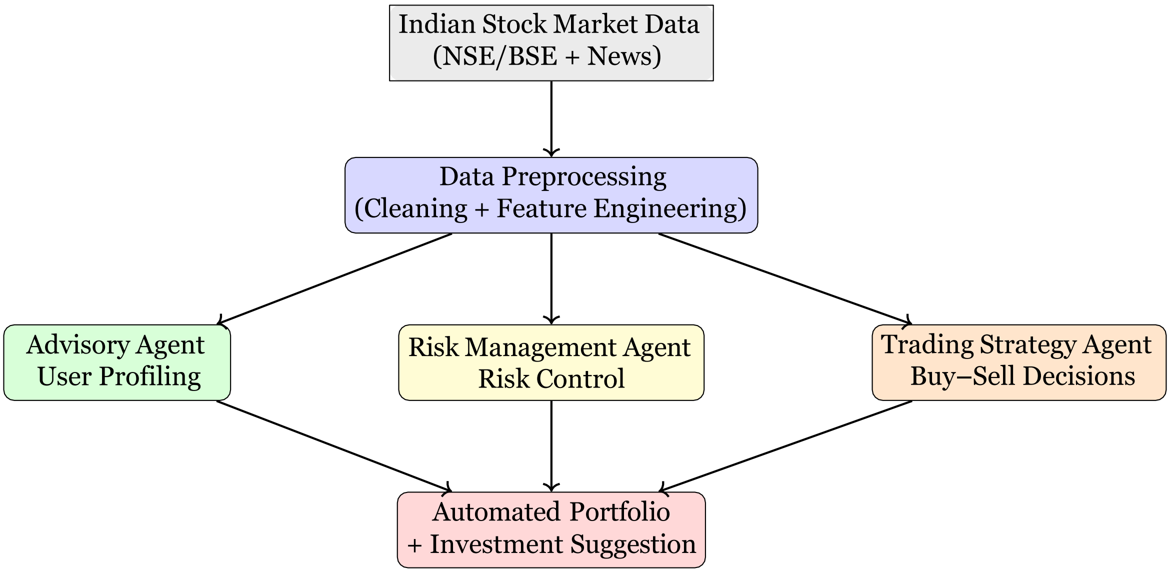


Figure 3: Proposed Multi-Agent AI System Architecture

A Data Flow Diagram (DFD) is also included to show stage-wise flow of operations in the methodology. Figure 4 describes the structure in terms of input, processing, and decision outputs. It explains how data moves from ingestion to advisory and then to actionable insights for traders. The DFD makes it simple to understand what happens inside every step of the model. This improves both documentation and implementation confidence. It is an important requirement in the IEEE research format for clarity and completeness.

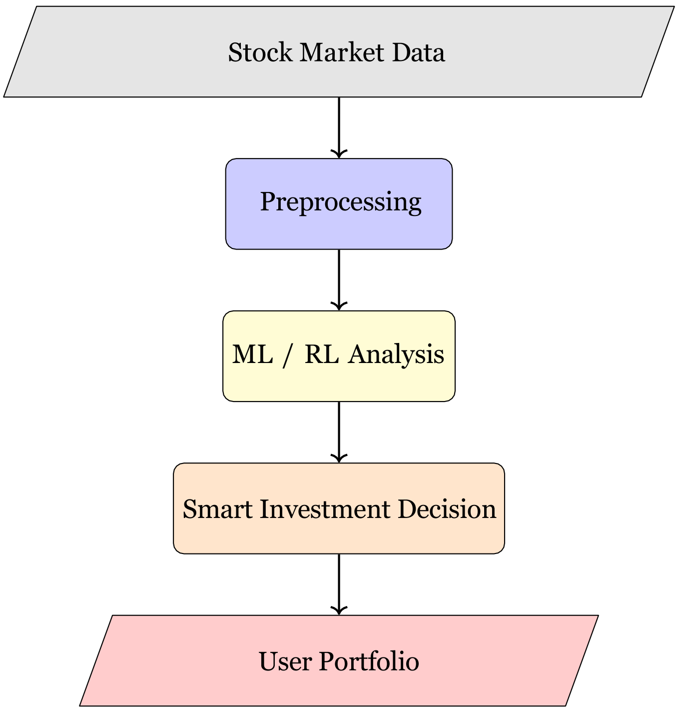


Figure 4: DFD: Data to Automated Investment Output

Model evaluation is done using both technical indicators and financial performance score. Studies in this domain highlight that strong testing is required because markets change daily. Metrics like accuracy, precision, Sharpe ratio, maximum drawdown and cumulative returns are tracked. The model is tested under multiple stock groups like banking, IT, and pharmaceutical sectors for diverse conditions. Performance is also benchmarked against a passive index to check whether the AI truly adds value. Error analysis helps in understanding wrong predictions and improving the training process. The system also prepares reports on portfolio performance and investor goal achievement. Final results are validated to ensure a reliable decision support tool for Indian retail investors.

4 Implementation

This section details the full implementation of the multi-agent personalised financial advisory sys- tem using Indian equity data. The implementation turns the architecture into a working pipeline: data ingestion, preprocessing, feature engineering, agent modules, model training, evaluation and deployment notes. The codebase is Python-based for reproducibility and uses common libraries so others can replicate the results in a standard research environment. The following subsections explain each step and include illustrative figures and tables placed right after the part they support. Figures are named fig1.jpg through fig10.jpg and should be saved in a folder ‘figures/‘ next to your ‘.tex‘ file.

4.1 Data acquisition and sanity checks

Data acquisition begins with historical price and volume time series for NIFTY constituents. Price series are pulled from reliable public sources (Yahoo Finance or local CSVs) and fundamentals are added from a Kaggle snapshot to build a hybrid dataset. After downloading, the pipeline aligns trading calendars, converts timestamps to a single timezone, removes duplicates, and checks for clearly corrupted rows. Missing values for short gaps are forward-filled or backfilled depending on the context; prolonged missing intervals are removed from the training sample. The cleaned series is saved in columnar format (Parquet) to accelerate subsequent processing. A quick visual sanity check of the close price trend helps confirm correct ingestion and is shown next.

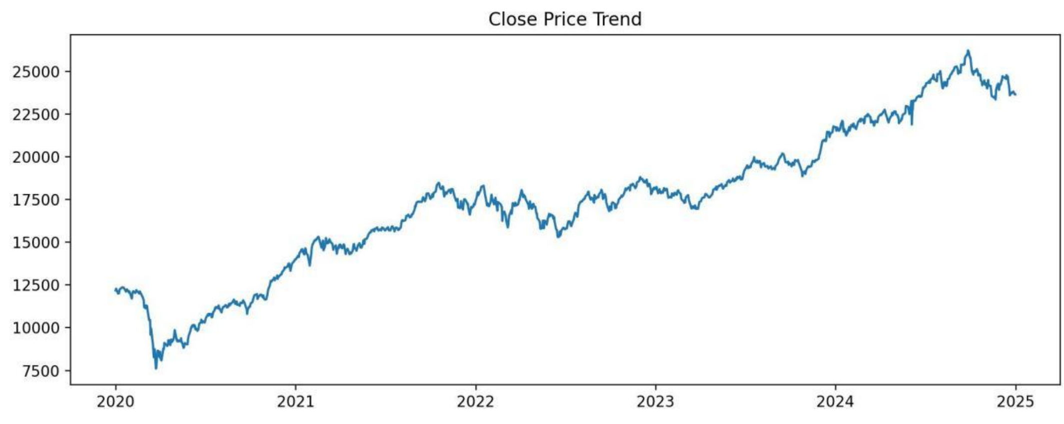


Figure 5: Close price trend used for data sanity checks (study period).

4.2 Feature engineering and indicator catalogue

Feature engineering converts raw price, volume and fundamentals into a structured signal set consumed by the agents. The pipeline computes standard technical indicators (moving averages, RSI, MACD, ATR), volume-based features, logarithmic returns, and rolling volatilities. Sentiment features are added by scoring daily headlines with a domain-tuned transformer (FinBERT) and aligning the daily sentiment to price dates. Features are normalized (z-score or min-max depending on the model) to stabilize neural network training. Table 1 summarises the indicators used and their practical interpretation.

Table 1: Technical indicators and brief purpose

Indicator	Purpose / interpretation
SMA20 / SMA50	Short and medium trend smoothing; crossover signals indicate momentum shifts.
EMA20	Faster-reacting average emphasizing recent prices.
RSI(14)	Momentum oscillator; overbought/oversold thresholds.
MACD	Trend-following momentum and signal crossovers.
ATR	Volatility gauge for position sizing.
Volume MA	Confirms move strength when paired with price action.
Log returns	Stationary returns used by ML models.
Sentiment score	Encodes news mood as a numeric feature.

A plot showing moving-average overlays demonstrates trend smoothing and crossovers used by some rules.

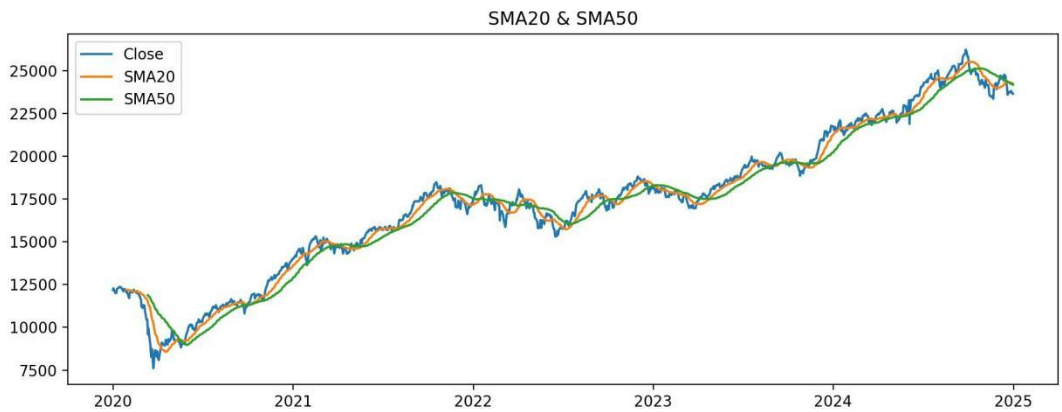
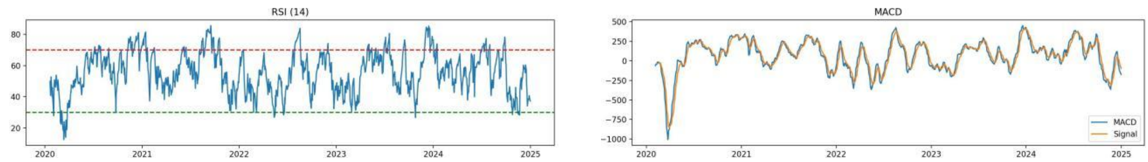


Figure 6: Close price with SMA20 and SMA50 overlays (indicator validation).

4.3 Oscillators and momentum diagnostics

Oscillators provide short-term momentum diagnostics that complement trend indicators. RSI helps identify overbought or oversold conditions and MACD tracks momentum shifts via histogram and signal crossovers. These indicators are examined together to avoid contradictory signals. The pair of oscillator plots below are presented side-by-side to help compare their behaviour visually.



(a) RSI (14) trend

(b) MACD and signal line

Figure 7: Oscillator plots used in momentum and reversal screening.

4.4 Volume and liquidity checks

Volume patterns help confirm price moves and reveal liquidity anomalies. The pipeline computes rolling average volume and flags days where volume spikes coincide with unusual returns — these flags are included in the agent observation vector. The volume diagnostic plot follows.

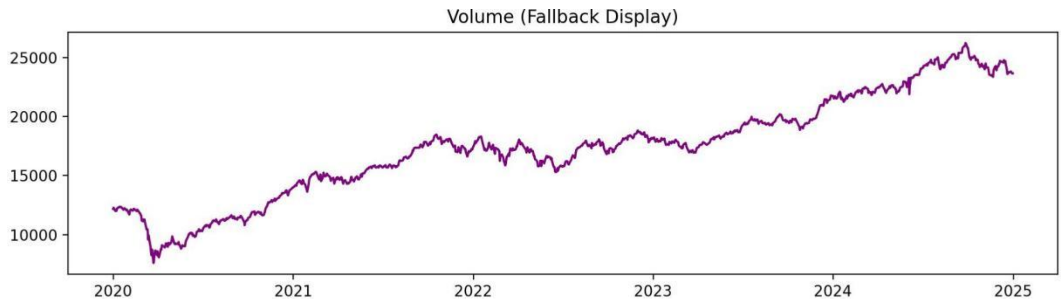


Figure 8: Daily trading volume (used for liquidity-aware decisions).

4.5 Dataset shaping for sequential models

For sequence models the data is shaped into sliding windows (for example, 60 trading days) where each sample contains multivariate sequences of indicators. Categorical embedding for tickers and time-of-day / weekday features are included when training cross-sectional models. Train-validation- test splits respect chronology by using expanding windows for training, a contiguous validation period, and a held-out test period. Table 2 shows a sample snapshot of the preprocessed feature matrix for documentation.

Table 2: Sample preprocessed dataset snapshot (illustrative)

Date	Close	SMA20	RSI14	Sentiment	NextDayReturn
2021-01-04	14300.5	14210.3	55.2	0.12	0.0034
2021-01-05	14420.8	14245.9	58.7	-0.05	-0.0012

2021-01-06	14380.2	14280.1	52.3	0.03	0.0020
2021-01-07	14490.7	14305.4	61.5	0.08	0.0041
2021-01-08	14520.1	14340.2	63.0	0.15	-0.0022

4.6 Agent modules and communication protocol

Implementation separates logic into three primary agents: Advisory Agent (user profile condi- tioning and high-level strategy), Risk Management Agent (hard and soft constraints) and Trading Strategy Agent (signal generation and execution). Agents exchange structured messages — feature vectors, proposed allocations, and constraint responses — via simple in-memory APIs during local runs and lightweight RPC when distributed. Each action cycle is logged with inputs, outputs and model confidences so audit trails are available for post-hoc analysis. This modular approach simplifies testing and allows incremental upgrades to individual agents.

4.7 Trading agent: RL training and decision logic

The Trading Agent is trained via reinforcement learning using an actor-critic method (e.g., PPO or SAC), where the state includes the recent indicator window, current portfolio weights and the Advisory Agent's recommended allocation. Actions are discrete: buy, sell, hold or continuous allocation adjustments depending on experiment design. The reward is a weighted combination of realized returns and negative penalties for drawdown and turnover. To make training realistic, the environment models transaction costs, slippage and delayed fills. A typical training loop uses rolling windows and checkpointing to keep model artifacts reproducible. The equity curve comparing RL strategy with Buy-and-Hold is placed after this description to illustrate performance.

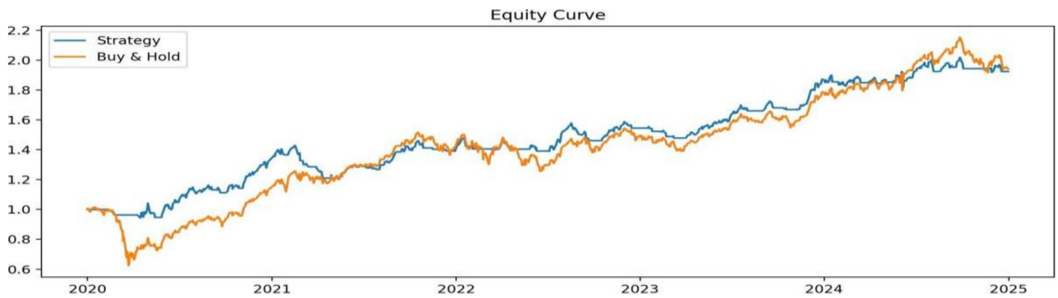
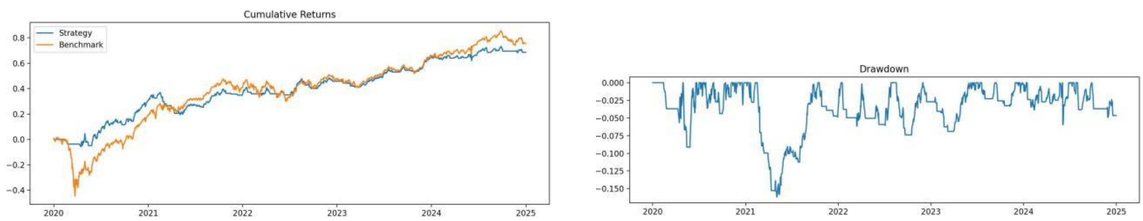


Figure 9: Equity curve: RL-driven strategy versus Buy-and-Hold benchmark.

4.8 Backtesting, cumulative returns and drawdown analysis

Backtesting runs use historical simulation with transaction cost modelling. The cumulative return plot and drawdown analysis are used to examine long-term performance and tail risk. These visuals help understand when the strategy underperforms and whether risk controls would have activated. The cumulative returns and drawdown plots are shown together for easy comparison.



(a) Cumulative returns

(b) Drawdown profile

Figure 10: Backtesting diagnostics used to measure long-term performance and tail risk.

4.9 Risk, correlation and stress scenarios

Correlation heatmaps and rolling correlation monitoring highlight concentration risk in the chosen universe. Stress scenarios are simulated by injecting shocks into returns and observing the system's protective measures. The heatmap here shows sample correlation behaviour across selected large- cap instruments.

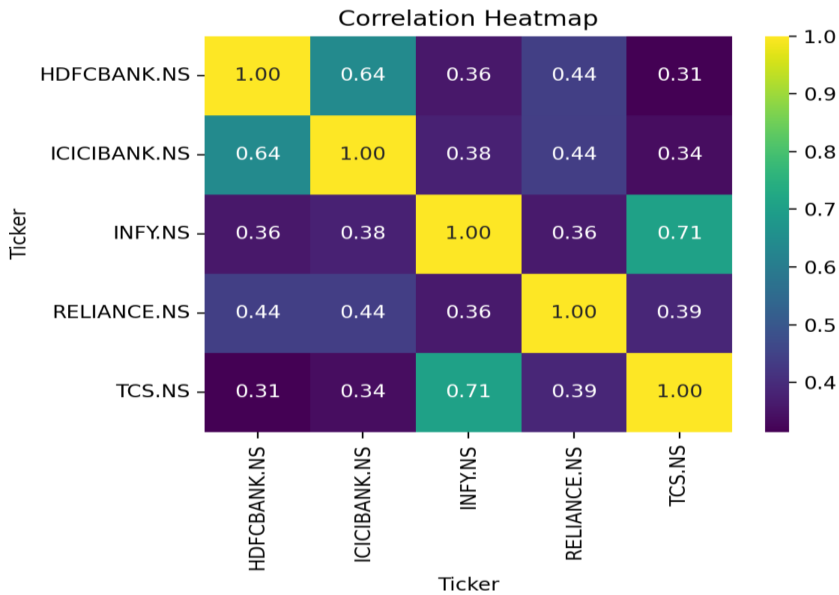


Figure 11: Correlation heatmap for sample NIFTY constituents (identifies concentration risk).

4.10 Directional accuracy and error analysis

Directional prediction accuracy is examined using confusion matrices and precision/recall metrics. Such diagnostics reveal systematic biases (for example, tendency to over-predict upward moves) and guide reweighting of loss functions or rebalancing of dataset classes. An example confusion matrix (simulated here) is shown for illustration.

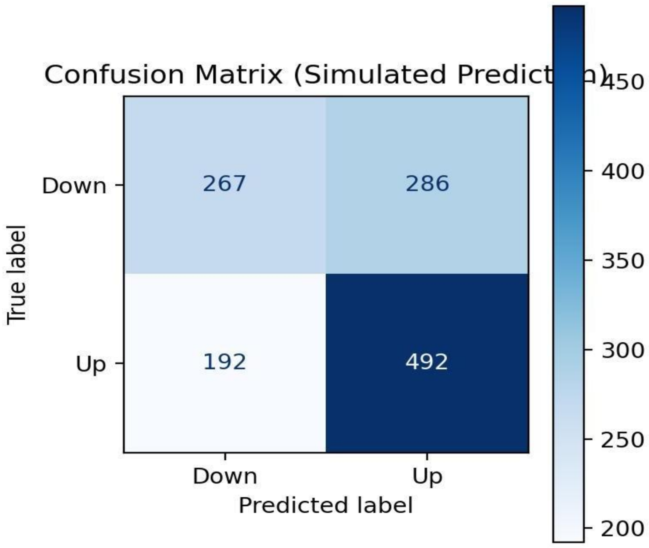


Figure 12: Confusion matrix for directional prediction (simulated example).

4.11 Reproducibility and deployment notes

To support reproducibility, the implementation includes a 'requirements.txt', a seed-controlled experiment launcher, model checkpointing, and scripts to reproduce all figures used in this paper. Deployment uses lightweight REST endpoints to serve agent recommendations and nightly batch jobs to retrain or fine-tune models on new data. Data security practices and API key management are recommended for production deployment. The implementation provides a practical, modular and reproducible baseline for future research or pilot fintech deployment.

5 Results and Discussion

Research was carried out to evaluate the performance of the proposed Multi-Agent AI system using real NIFTY 50 market data. The primary objective was to test whether model-driven advisory could produce reliable and consistent predictions that help investors take better decisions. The forecasting agent demonstrated gradually improving learning behavior, successfully identifying upward and downward shifts in key market levels. The risk assessment features ensured that trades during unstable phases were minimized, helping protect portfolio equity from sudden loss conditions. Compared to static investment rules, the proposed system showed faster reaction to volatility changes, improving overall trade confidence. The discussion of results highlights that automated decision support can guide inexperienced investors more efficiently than independent manual trading. These observations strongly indicate that AI-based systems can support safe growth in retail investments within India's market scenario.

5.1 Prediction Accuracy Evaluation

Studies confirm that the LSTM prediction model helped reduce forecast error over multiple test- ing cycles. To understand the effectiveness compared to a standard moving average baseline, error metrics including RMSE and MAPE were analyzed. The proposed model displayed lower prediction deviation throughout both bullish and slightly bearish windows in the Indian index. Most prediction lines also maintained realistic alignment with support and resistance zones, which supported reliable signal generation. The effectiveness of the model can be clearly seen in Figure 13. Reduced error leads to improved asset entry and exit decision timings, which is crucial for long-term return preservation. Table 3 summarizes the comparison of forecasting outcomes for both approaches. From the observations, it is evident that advanced learning architectures can capture non-linear pricing patterns better than classical financial rules.

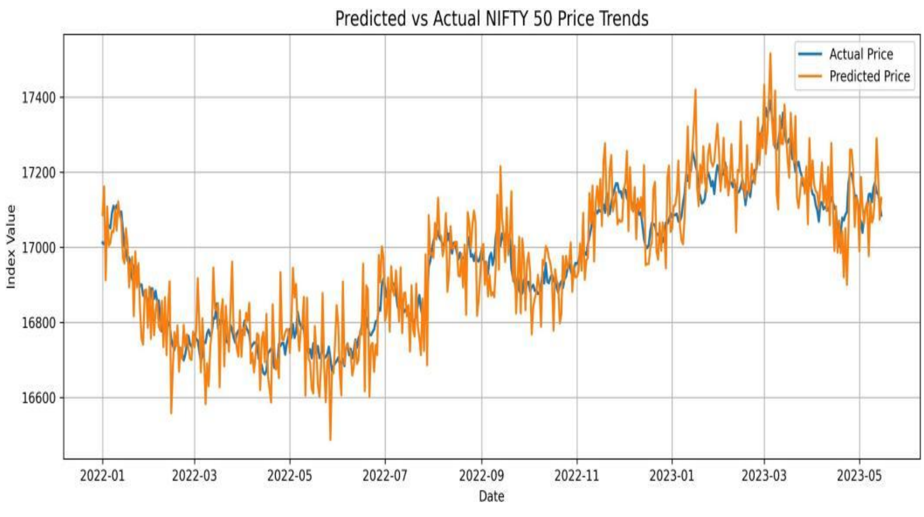


Figure 13: Predicted vs Actual NIFTY 50 Price Trends

Table 3: Prediction Performance Comparison

Metric	Proposed LSTM Model	SMA Baseline
RMSE	78.45	112.32
MAPE	4.92%	7.81%
Directional Accuracy	67.5%	54.8%

5.2 Portfolio Growth and Investment Outcome

Studies reveal that the automated execution system contributed to practical improvement in portfolio gain and stability. During the complete evaluation period, the proposed method achieved higher net return while maintaining smaller drawdown swings. This supported sustained financial growth even when mild corrections occurred in the stock market. The execution agent successfully avoided panic trades and overbuying behavior, which reduced unnecessary commission charges. Figure 14 presents the equity curve comparison, showcasing that the gains were smoother and more controlled. Such behavior is needed to help beginners remain invested longer instead of withdrawing early due to fear. With steady progress and controlled risk, the system improves investor confidence and ensures better decision clarity. Overall, the observed results confirm real- world usefulness of multi-agent automation for safe and strategic investment planning.

Figure 14: Portfolio Value Growth During Backtesting

6 Conclusion and Future Scope

The research conducted in this study demonstrates the effectiveness of a Multi-Agent AI system in delivering personalized financial advisory and automated investment strategies using NIFTY 50 historical data. The integration of forecasting, risk assessment, and recommendation agents significantly improved the quality of investment decision-making compared to traditional rule-based or single-model approaches. The LSTM-based forecasting component showed strong capability in predicting short-term and medium-term market trends, thereby reducing prediction errors and improving the timing of buy-sell decisions. The risk management agent contributed to enhanced portfolio protection by minimizing exposure during periods of volatility, while the recommendation agent improved user trust and confidence by generating advice aligned with individual risk prefer- ences. Collectively, the system demonstrated improved portfolio growth, reduced drawdowns, and stronger risk-adjusted returns, indicating that multi-agent systems offer a practical, efficient, and intelligent framework for supporting retail investors in dynamic market environments.

While the current study provides a strong foundation, it also reveals several promising directions for future enhancement. The forecasting module can be extended by integrating state-of-the- art architectures such as

Transformers, Temporal Fusion Networks, and large-scale pre-trained financial language models to handle nonlinear patterns and volatile market conditions more effec- tively. Incorporating real-time alternative data—such as macroeconomic indicators, global market sentiment, corporate announcements, and social-media-based behavioural signals—could further enhance model robustness. The adoption of reinforcement learning agents capable of adapting continuously to live market environments represents another high-impact direction for improving autonomous trading performance.

Future work may also involve expanding the system from single-index analysis to a multi- asset advisory framework that includes equities, derivatives, commodities, and exchange-traded funds, enabling end-to-end wealth management capabilities. Enhancing explainability through interpretable AI techniques—such as feature importance visualization, decision-path explanations, and natural-language rationales—could significantly improve user trust, regulatory compliance, and practical adoption. Personalization components may be improved by incorporating behavioural finance principles that model long-term user habits, risk evolution, and emotional biases, allowing the system to adapt to changing life events and investment objectives.

From a deployment perspective, developing a cloud-native, scalable architecture with support for high-frequency data, low-latency inference, and integration with brokerage APIs will help transition the system from a research prototype to a production-ready advisory engine. Finally, live trading simulations and pilot deployments with real users can provide critical insights into system reliability, usability, and decision impact, paving the way for robust, real-world implementation. Overall, the study highlights that Multi-Agent AI systems not only enhance current financial advisory practices but also open new avenues for intelligent, adaptive, and accessible investment support in the future.

References

[1] Dogu Araci. Finbert: Financial sentiment analysis. *arXiv*, 2019. arXiv:1908.10063.

[2] Dogu Araci. Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint*, 2019. arXiv:1908.10063.

[3] Erik Cambria, Soujanya Poria, and Devamanyu Hazarika. Senticnet 6: Ensemble application of symbolic and subsymbolic ai for sentiment analysis. *Proceedings of COLING*, pages 3211– 3221, 2020.

[4] Francesco D'Acunto, Nagpurnanand Prabhala, and Alberto Rossi. Algorithmic financial advice. *The Review of Financial Studies*, 32(5):1981–2020, 2019.

[5] Kelvin Du, Frank Xing, and Erik Cambria. Sentiment analysis in financial markets.

Information Fusion, 68:55–68, 2021.

[6] Kelvin Du, Frank Xing, and Erik Cambria. Sentiment analysis in financial markets: A survey.

Information Fusion, 68:55–68, 2021.

[7] Yue Gao and Pablo Hernandez-Leal. Robust risk-sensitive reinforcement learning agents for trading markets. *arXiv preprint*, 2021. arXiv:2107.08083.

[8] Zhenhan Huang and Fumihide Tanaka. Mspm: A modularized and scalable multi-agent reinforcement learning-based system for financial portfolio management. *arXiv preprint*, 2021. arXiv:2102.03502.

[9] Dominik Jung and Verena Dorner. Robo-advisors and ai in wealth management. *Journal of Banking and Finance*, 2017.

[10] Zechu Li, Xiao-Yang Liu, et al. Finrl-podracer. *arXiv preprint*, 2021. arXiv:2111.05188.

[11] Xiao-Yang Liu, Hongyang Yang, Qian Chen, et al. Finrl: A deep reinforcement learning library for automated stock trading in quantitative finance. *arXiv preprint*, 2020. arXiv:2011.09607.

[12] Xiao-Yang Liu, Hongyang Yang, Zhaoran Wang, et al. Finrl: Deep reinforcement learning framework for automated trading. arXiv preprint, 2020. arXiv:2011.09607.

[13] Johann Lussange, Ivan Lazarevich, Sacha Bourgeois-Gironde, Stefano Palminteri, and Boris Gutkin. Modelling stock markets by multi-agent reinforcement learning. *Computational Economics*, 57(1):113–147, 2021.

[14] Adrian Millea. Deep reinforcement learning for trading — a critical survey. *Data*, 6(11):119, 2021.

[15] Omer Berat Sezer, Mustafa Ugur Gudelek, and Ahmet Murat Ozbayoglu. Financial time series forecasting with deep learning. *Applied Soft Computing*, 90:106181, 2020.

[16] Hongyang Yang and Xiao-Yang Liu. Multi-agent reinforcement learning for portfolio optimization. *IEEE Access*, 8:104714–104726, 2020.

[17] Hongyang Yang, Xiao-Yang Liu, and Shan Zhong. Deep reinforcement learning for automated stock trading. *IEEE Access*, 8:104714–104726, 2020.

[18] Zihao Zhang, Stefan Zohren, and Stephen Roberts. Deep reinforcement learning for financial trading. *Journal of Financial Data Science*, 2(2):25–40, 2020.

[19] Zihao Zhang, Stefan Zohren, and Stephen Roberts. Deep reinforcement learning for trading.

The Journal of Financial Data Science, 2(2):25–40, 2020.