A Multi-Agent Financial Intelligence Framework for Financial Decision Making

Enjamul Islam Khan[1], Alapan Kuila[1]

[1]*Indian Institute of Information Technology Design and Manufacturing Kurnool, Kurnool, Andhra Pradesh, India*

Abstract

Financial decision making remains a challenging problem because modern financial markets generate heterogeneous information, including historical market prices, financial news, technical indicators, corporate filings, and rapidly evolving market events. Effectively integrating these diverse information sources into reliable BUY, HOLD, or SELL recommendations under real-time deployment constraints is a key objective of the CLEF 2026 FinMMEval Task 3 benchmark. This paper presents a modular Multi-Agent Financial Intelligence Framework that combines complementary reasoning obtained from financial sentiment analysis, technical analysis, event detection, filing analysis, financial semantic understanding, and multi-horizon quantitative forecasting. Each specialized reasoning agent independently analyzes one aspect of the financial environment and produces a trading recommendation together with an associated confidence score. The outputs of all reasoning agents are subsequently integrated using a supervised meta-classification model to generate the final trading decision. The complete system is deployed as a lightweight FastAPI service for real-time financial decision support. Experimental evaluation through internal backtesting demonstrates that the proposed framework achieves strong cumulative return, competitive risk-adjusted performance, and consistent improvements over individual forecasting models and conventional static ensemble approaches. These results demonstrate the effectiveness of combining heterogeneous financial reasoning within a unified multi-agent decision-making framework.

1. Introduction

Financial decision making remains one of the most challenging problems in artificial intelligence because financial markets continuously evolve under uncertain and rapidly changing conditions [1, 2]. Modern financial environments generate heterogeneous information, including historical market prices, financial news, technical indicators, corporate filings, macroeconomic events, and investor sentiment [3, 4]. Effectively combining these diverse information sources into reliable BUY, HOLD, or SELL recommendations remains a challenging research problem.

Recent advances in financial natural language processing and machine learning have substantially improved financial forecasting and market understanding. Domain-specific language models such as FinBERT [5] and FinGPT [6] have demonstrated strong performance in financial sentiment analysis, while machine learning and deep learning models, including XGBoost [7], LSTM networks [8], and Transformer-based architectures [9, 10, 11], have achieved promising results for financial time-series forecasting. Nevertheless, many existing approaches rely on a single predictive model or focus exclusively on price forecasting, limiting their ability to integrate heterogeneous financial information for practical financial decision support [12].

This work was developed for CLEF 2026 FinMMEval Task 3: Financial Decision Making [13, 14, 15], which evaluates reasoning-to-action financial intelligence systems under realistic deployment constraints. Unlike conventional forecasting benchmarks, participating systems must process structured daily financial contexts and generate actionable BUY, HOLD, or SELL recommendations together with concise natural-language explanations.

To address these challenges, we propose a modular multi-agent financial intelligence framework that combines complementary reasoning obtained from financial news, technical indicators, event detection, corporate filings, financial semantic analysis, and multi-horizon quantitative forecasting. Each specialized reasoning agent independently analyzes one aspect of the financial environment and produces a trading recommendation with an associated confidence score. The outputs of these agents

CLEF 2026 Working Notes, 21–24 September 2026, Jena, Germany

are subsequently integrated through a supervised meta-classification model that generates the final trading decision using ensemble learning principles [16].

The proposed system is designed for practical real-time financial decision support and is deployed as a lightweight FastAPI service compatible with the FinMMEval Task 3 evaluation protocol. Experimental results demonstrate that integrating heterogeneous financial reasoning agents improves cumulative return and risk-adjusted performance compared with individual forecasting models and conventional static ensemble approaches [17].

2. Related Work

Recent advances in artificial intelligence have significantly improved financial forecasting, financial language understanding, and automated decision support. Transformer-based language models such as BERT [18] have substantially improved contextual representation learning and inspired domain-specific financial models including FinBERT [5], FinGPT [6], and BloombergGPT [19], which have significantly advanced financial sentiment analysis and financial reasoning.

Financial market prediction has also been extensively studied using machine learning and deep learning techniques. Traditional machine learning approaches, including XGBoost [7], Random Forests [20], and LightGBM [21], have demonstrated strong predictive performance on structured financial data and technical indicators. Deep learning models based on LSTM [8], GRU [22], CNN-LSTM hybrids [23], Transformer architectures [9], Temporal Fusion Transformers [10], and PatchTST [11] have further improved financial time-series forecasting by modeling complex temporal dependencies and nonlinear market dynamics [24]. However, these approaches typically rely on a single predictive model and primarily focus on forecasting future prices rather than supporting complete financial decision making [12].

Recent research has increasingly explored ensemble learning and multi-source information fusion for financial prediction. Ensemble approaches combine multiple predictive models to improve robustness and reduce the limitations of individual forecasting techniques [17, 25, 16]. Nevertheless, many existing ensemble methods employ fixed combination strategies or static model aggregation, making them less suitable for financial environments where heterogeneous textual and numerical information must be jointly considered during decision making [26].

Unlike conventional forecasting systems, FinMMEval Task 3 requires complete reasoning-to-action financial intelligence, where systems must integrate heterogeneous financial information and generate actionable BUY, HOLD, or SELL recommendations together with concise natural-language explanations under real-time deployment constraints [13, 14, 15]. These requirements extend beyond conventional price forecasting and require the integration of financial sentiment, technical indicators, event information, corporate filings, and quantitative forecasting within a unified decision-making framework.

Motivated by these challenges, the proposed framework combines multiple specialized financial reasoning agents with a supervised meta-classification model that integrates complementary information from textual, numerical, technical, and event-driven financial sources. Rather than relying on a single forecasting model, the proposed architecture performs modular financial reasoning followed by meta-level decision integration using stacked ensemble learning principles [16], providing a practical framework for real-time financial decision support under the FinMMEval Task 3 benchmark.

3. FinMMEval Task 3: Problem Definition

FinMMEval 2026 Task 3 focuses on financial decision making under realistic reasoning-to-action trading scenarios [13, 14, 15]. For each evaluation instance, the system receives a structured daily JSON-based financial context containing historical market prices, financial news articles, momentum annotations, and optional SEC 10-K/10-Q corporate filings. Based on this heterogeneous financial information, the objective is to generate both an actionable trading recommendation and a concise natural-language explanation.

Formally, each daily financial context can be represented as

$$C = \{P_t, N_t, M_t, F_t\}, \tag{1}$$

where P_t denotes the historical market information, N_t represents the collection of financial news articles, M_t denotes the momentum annotations, and F_t represents the optional corporate filing information. Given the financial context C, the system predicts one trading action from the set

$$A = \{\text{BUY}, \text{HOLD}, \text{SELL}\}. \tag{2}$$

Unlike traditional financial forecasting benchmarks that focus only on price prediction, FinMMEval Task 3 evaluates complete decision-making systems capable of integrating heterogeneous numerical and textual financial information to generate practical trading recommendations. In addition to prediction accuracy, submitted systems must satisfy real-time inference constraints and produce concise evidence-based rationales for their recommendations. These requirements motivate the proposed multi-agent framework, which combines specialized financial reasoning agents with a supervised meta-classification model to generate robust BUY, HOLD, or SELL decisions.

4. Proposed Framework

The proposed framework is a modular multi-agent financial intelligence system designed to generate BUY, HOLD, or SELL recommendations from heterogeneous financial information. Instead of relying on a single forecasting model, the framework combines complementary reasoning obtained from financial news, historical market prices, technical indicators, financial events, corporate filings, and quantitative forecasting. The complete decision-making process consists of three stages: (i) financial context processing, (ii) specialized multi-agent reasoning, and (iii) meta-agent decision integration.

4.1. Financial Context Processing

The framework receives a daily financial context consisting of historical market prices, financial news articles, momentum annotations, historical price trajectories, and optional SEC 10-K/10-Q corporate filings provided by the FinMMEval Task 3 benchmark. Since these information sources contain both structured numerical data and unstructured financial text, a preprocessing stage is first applied to normalize the incoming data and remove incomplete records.

Following preprocessing, a feature engineering module extracts quantitative market features from historical prices. The generated feature set includes technical indicators such as Relative Strength Index (RSI), Moving Average Convergence Divergence (MACD), Simple Moving Averages (SMA), Exponential Moving Averages (EMA), Average True Range (ATR), Bollinger Bands, historical returns, volatility measurements, and trend-related features. These engineered variables provide a structured representation of the current market state and serve as inputs to the quantitative reasoning components.

4.2. Multi-Agent Financial Reasoning

After preprocessing, the financial context is analyzed in parallel by six specialized reasoning agents, each focusing on a different aspect of the financial environment.

The **News Sentiment Agent** employs the domain-specific FinBERT model to estimate financial sentiment from daily news articles. The **Technical Analysis Agent** performs rule-based market analysis using technical indicators including RSI, MACD, moving averages, Bollinger Bands, ATR, volatility, and trend-strength measurements. The **Event Detection Agent** identifies financially significant events using predefined bullish and bearish financial event patterns extracted from news articles.

To incorporate longer-term financial information, the **Filing Analysis Agent** analyzes optional SEC 10-K and 10-Q reports containing corporate fundamentals and forward-looking business information. The **Financial Semantic Agent** combines FinBERT-based financial language understanding with domain-specific financial keyword refinement to capture contextual financial meaning beyond simple sentiment polarity. Finally, the **Multi-Horizon Quantitative Agent** performs quantitative forecasting using three independent XGBoost classification models corresponding to 1-day, 3-day, and 5-day prediction horizons.

Each reasoning agent independently produces a BUY, HOLD, or SELL recommendation together with an associated confidence score, providing complementary evidence for the final decision-making stage.

4.3. Meta-Agent Decision Integration

The outputs generated by all specialized reasoning agents are combined with the engineered market features to construct a unified feature representation. This feature vector is provided as input to a supervised meta-classification model that learns how to combine complementary reasoning signals from textual, numerical, technical, and event-driven financial information.

During inference, the meta-classifier predicts the final BUY, HOLD, or SELL recommendation together with the corresponding class probabilities. The maximum predicted class probability is used as the confidence score of the final recommendation. To improve operational robustness, a lightweight safety verification stage is subsequently applied to identify low-confidence predictions and highly uncertain market conditions. Under such situations, the framework conservatively defaults to HOLD recommendations, reducing unnecessary trading risk during real-time deployment.

The final output of the framework consists of the recommended trading action, its associated confidence score, and a concise natural-language rationale summarizing the primary financial evidence supporting the decision. The complete architecture enables complementary reasoning across heterogeneous financial information sources while maintaining an efficient and interpretable decision-making process suitable for practical financial decision support.

The proposed framework transforms heterogeneous financial information into actionable trading decisions through a modular multi-agent architecture. Instead of relying on a single prediction model, the system combines multiple specialized reasoning agents that independently analyze complementary aspects of the financial environment, following the principles of modular and ensemble intelligence [17, 16]. The architecture is designed to process both structured numerical market data and unstructured financial text while supporting efficient real-time inference through a lightweight API-based deployment using FastAPI [27].

The overall pipeline begins by receiving the daily financial context, including historical market prices, financial news articles, momentum annotations, and optional corporate regulatory filings (SEC 10-K and 10-Q reports). The incoming data are first preprocessed to remove incomplete records, normalize heterogeneous data formats, and prepare both numerical and textual information for downstream analysis. A feature engineering stage subsequently extracts quantitative market features, including technical indicators such as RSI, MACD, Bollinger Bands, and ATR [28, 29, 30], together with volatility measurements, trend characteristics, and other derived financial variables used throughout the reasoning pipeline.

After preprocessing, the financial context is analyzed in parallel by six specialized reasoning agents: (1) the News Sentiment Agent, (2) the Technical Analysis Agent, (3) the Event Detection Agent, (4) the Filing Analysis Agent, (5) the Financial Semantic Agent, and (6) the Multi-Horizon Quantitative Agent. Each agent independently evaluates a specific aspect of the financial environment and produces both a trading recommendation (BUY, HOLD, or SELL) and an associated confidence score. This modular design enables the framework to capture complementary information from textual, numerical, event-driven, and quantitative financial sources.

The outputs of all reasoning agents are combined with the engineered market features to construct a unified feature representation. This feature vector is provided as input to a supervised meta-classification model based on XGBoost [7], which predicts the final BUY, HOLD, or SELL recommendation together with the corresponding class probabilities. The prediction confidence is subsequently used by a lightweight post-processing module that performs safety verification under uncertain market conditions. Predictions associated with low confidence or abnormal market behaviour are conservatively converted

Algorithm 1 Multi-Agent Financial Decision Pipeline

Require: Daily financial context C
Ensure: Final trading recommendation $A \in \{\text{BUY}, \text{HOLD}, \text{SELL}\}$
Receive daily financial context containing historical prices, financial news, momentum information, and optional SEC filings.
Perform data preprocessing and normalize heterogeneous financial inputs.
Extract engineered financial features including RSI, MACD, SMA, EMA, ATR, Bollinger Bands, volatility, and trend-related indicators.
Execute the News Sentiment Agent to analyze financial news.
Execute the Technical Analysis Agent using engineered market indicators.
Execute the Event Detection Agent to identify market-moving financial events.
Execute the Filing Analysis Agent to analyze institutional filings when available.
Execute the Financial Semantic Agent to capture contextual financial semantics.
Execute the Multi-Horizon Quantitative Agent using the 1-day, 3-day, and 5-day forecasting models.
Construct the meta-level feature vector by combining agent outputs with engineered technical features.
Predict the final trading action using the XGBoost meta-classifier.
Estimate prediction confidence from the meta-classifier output probabilities.
if prediction confidence is low or abnormal market conditions are detected **then**
    Convert recommendation to HOLD.
end if
Generate a concise natural-language rationale.
Return the final BUY, HOLD, or SELL recommendation together with the explanation.

to HOLD recommendations to improve decision stability during real-time deployment.

Finally, the framework generates a structured API response consisting of the recommended trading action, the associated confidence score, and a concise natural-language rationale summarizing the primary financial evidence supporting the recommendation. The modular architecture enables efficient integration of heterogeneous financial information while maintaining a unified and interpretable decision-making process suitable for practical financial decision support.

The overall architecture of the proposed Multi-Agent Financial Intelligence Framework is illustrated in Figure 1.

Algorithm 1 summarizes the complete reasoning pipeline of the proposed framework, from financial context preprocessing to final trading recommendation generation.

5. Heterogeneous Financial Signal Processing

The proposed framework analyzes financial markets through multiple specialized reasoning agents, where each agent focuses on a distinct source of financial information. Instead of relying on a single forecasting model, the framework decomposes the decision-making process into complementary reasoning modules that analyze financial news, historical market data, technical indicators, corporate filings, financial events, and multi-horizon market dynamics. Each agent independently generates a trading recommendation (BUY, HOLD, or SELL) together with an associated confidence score, which is subsequently integrated by the Meta classification Layer described in Section 6.

5.1. News Sentiment Agent

The News Sentiment Agent analyzes daily financial news using the domain-specific FinBERT model [5]. Given a collection of news articles associated with a financial asset, each article is processed independently to estimate positive, neutral, and negative sentiment probabilities. Following the implementation, the sentiment score is computed as the difference between the positive and negative class probabilities:

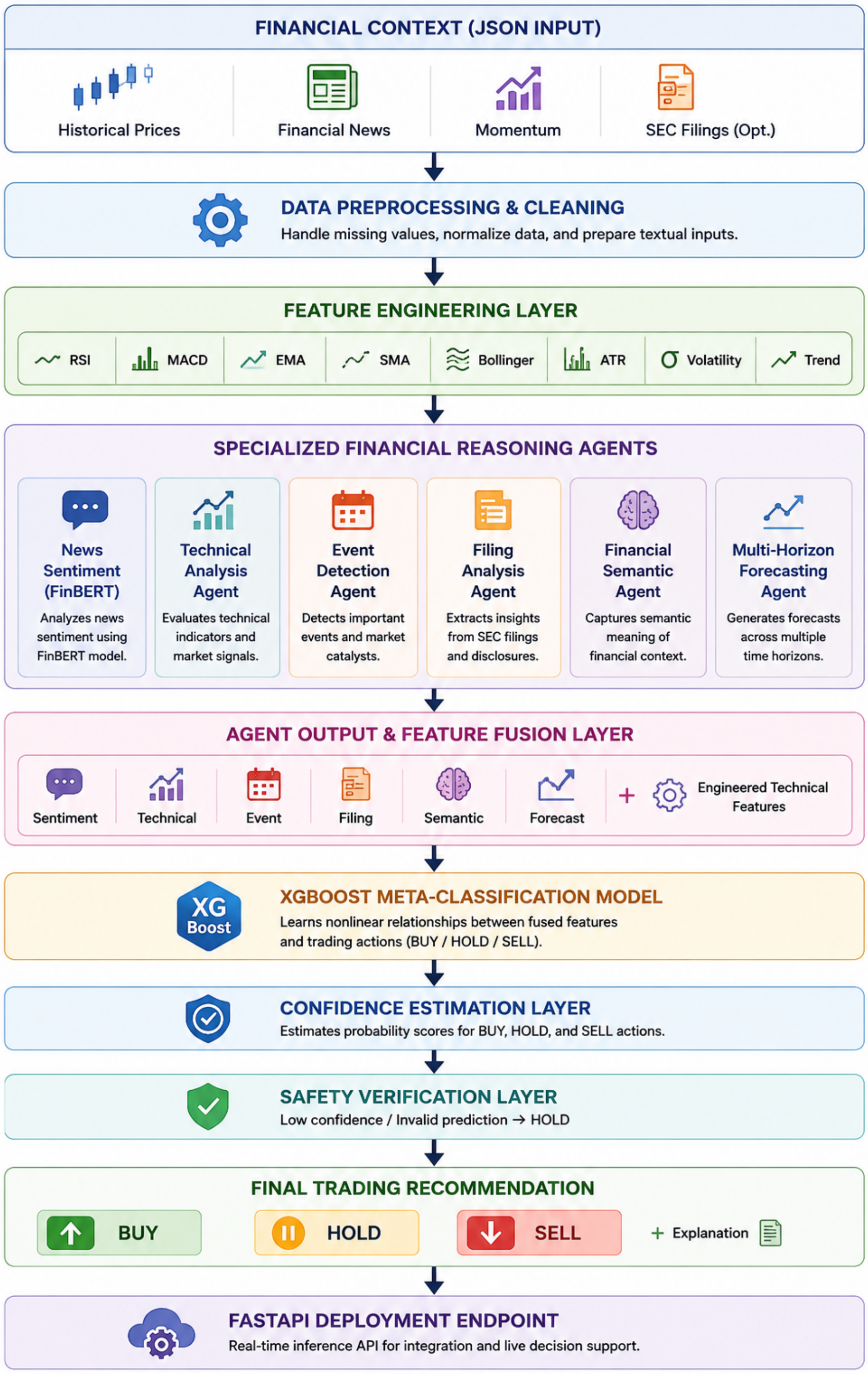

Figure 1: Overall architecture of the proposed Multi-Agent Financial Intelligence Framework. Daily financial contexts are preprocessed and transformed into engineered market features, which are analyzed by six specialized reasoning agents. Their outputs are fused and passed to an XGBoost-based meta-classification model that generates the final BUY, HOLD, or SELL recommendation together with a confidence score. A lightweight safety verification layer improves the robustness of real-time deployment.

$$s_i = P_{\text{positive}} - P_{\text{negative}} \tag{3}$$

where P_{positive} and P_{negative} denote the probabilities predicted by FinBERT for the corresponding news article. The final sentiment score is obtained by averaging the sentiment scores across all available financial news:

$$S_{\text{news}} = \frac{1}{N}\sum_{i=1}^{N} s_i \tag{4}$$

where N represents the number of news articles. Based on the aggregated sentiment score, the agent produces a BUY, HOLD, or SELL recommendation together with a confidence score that is forwarded to the meta-agent.

5.2. Technical Analysis Agent

The Technical Analysis Agent performs rule-based market analysis using historical price information and multiple technical indicators. The implementation extracts quantitative features including Relative Strength Index (RSI), Moving Average Convergence Divergence (MACD), Simple Moving Averages (SMA), Exponential Moving Averages (EMA), Average True Range (ATR), Bollinger Bands, market volatility, and trend-strength indicators.

The agent combines these indicators through predefined decision rules to identify momentum reversals, trend continuation, and overbought or oversold market conditions. For example, RSI and moving-average relationships are jointly used to identify potential BUY or SELL opportunities, while MACD and volatility measurements provide additional confirmation of market direction. The agent outputs both a trading recommendation and an associated confidence score for subsequent ensemble integration.

5.3. Event Detection Agent

The Event Detection Agent identifies financially significant events directly from textual news content using a lightweight rule-based event extraction strategy. Instead of employing a large language model, the implementation maintains predefined collections of bullish and bearish financial event keywords covering earnings announcements, guidance revisions, mergers and acquisitions, regulatory actions, lawsuits, fraud reports, layoffs, bankruptcy events, and other market-moving developments.

The occurrence of bullish and bearish events is counted within the incoming financial news, and the resulting event statistics are converted into BUY, HOLD, or SELL recommendations using deterministic decision rules. The confidence assigned to each prediction increases with the consistency and frequency of detected financial events. This lightweight design enables efficient real-time inference while maintaining transparent and reproducible event reasoning.

5.4. Filing Analysis Agent

The Filing Analysis Agent analyzes optional corporate regulatory filings, including SEC 10-K and SEC 10-Q reports, to capture longer-term fundamental information that is not immediately reflected in short-term market movements. The module focuses on financial statements, business outlook, operational risks, revenue trends, debt exposure, and forward-looking management discussions.

The extracted filing information provides complementary institutional signals that improve long-term financial reasoning, particularly when short-term news sentiment alone is insufficient for reliable investment decisions.

5.5. Multi-Horizon Quantitative Agent

The Multi-Horizon Quantitative Agent performs quantitative market forecasting using three independent XGBoost classification models [7], corresponding to prediction horizons of one day, three days, and five days. Each model independently estimates the probabilities of BUY, HOLD, and SELL actions using engineered market features extracted from historical price data.

Rather than relying on a single prediction horizon, the framework combines the probabilistic outputs of the three forecasting models using weighted probability fusion, where larger weights are assigned to longer prediction horizons to improve decision stability. The resulting probability distribution provides the quantitative forecasting component supplied to the Meta Classification Layer.

The selection of 1-day, 3-day, and 5-day forecasting horizons enables the framework to capture complementary market behavior at different temporal scales. The 1-day model focuses on immediate market reactions to breaking financial news and short-term momentum changes. The 3-day model captures short-term trend evolution after information diffusion and delayed investor responses, while the 5-day model models broader weekly market movements that are generally less sensitive to daily price noise. Combining these complementary forecasting horizons improves temporal consistency and provides a more robust quantitative representation than relying on a single prediction horizon.

5.6. Financial Semantic Agent

The Financial Semantic Agent (FinGPT) performs semantic financial reasoning beyond conventional sentiment analysis by leveraging a financial large language model. While the News Sentiment Agent estimates overall market sentiment, the Financial Semantic Agent captures deeper contextual financial information, including earnings expectations, institutional confidence, strategic partnerships, regulatory developments, and financial risk factors.

The agent analyzes the semantic meaning of financial narratives and generates a BUY, HOLD, or SELL recommendation together with an associated confidence score. This complementary semantic reasoning enables the framework to distinguish between financially similar news articles that may imply different investment decisions. The resulting prediction and confidence score are subsequently incorporated into the Meta Classification Layer.

6. Meta-Agent Ensemble Learning

The final financial recommendation is generated through a meta-agent ensemble that integrates the outputs of multiple specialized reasoning agents into a unified decision model. Rather than relying on a single forecasting component, the proposed framework combines complementary information obtained from the News Sentiment Agent, Technical Analysis Agent, Event Detection Agent, Filing Analysis Agent, Financial Semantic Agent, and Multi-Horizon Quantitative Agent. Each agent independently analyzes a different aspect of the financial environment and produces an intermediate trading signal that is subsequently fused for final decision making.

The outputs of the specialized reasoning agents are combined with market-specific quantitative features extracted during the feature engineering stage. These include Relative Strength Index (RSI), Moving Average Convergence Divergence (MACD), volatility measures, trend-related indicators, Bollinger Band characteristics, and other engineered financial features. The resulting feature vector captures both the opinions of the specialized agents and the current market state, providing a comprehensive representation for the final decision model.

6.1. Meta-Classifier Training

Instead of manually assigning fixed importance weights to individual reasoning agents, the proposed framework employs a supervised XGBoost-based meta-classifier to learn the relationship between heterogeneous financial signals and the final trading decision.

During training, a meta-level feature vector is constructed by combining the outputs of the specialized reasoning agents with engineered market features. Based on the implementation, the meta-classifier receives a 16-dimensional feature vector consisting of five agent actions (News Sentiment, FinGPT, Event Detection, Technical Analysis, and Multi-Horizon Quantitative), five corresponding confidence scores, and six engineered market features including the Relative Strength Index (RSI), Moving Average Convergence Divergence (MACD), volatility, trend strength, Bollinger Band width, and market regime. The Filing Analysis Agent operates as an optional reasoning component because SEC filings are not available for every financial context in the FinMMEval Task 3 dataset. Consequently, filing-based information contributes only when such documents are available and is not included in the fixed-length meta-classifier feature vector.

To improve learning under class imbalance, class-specific sample weights are applied during training, assigning higher importance to BUY and SELL samples than HOLD samples. The meta-classifier is implemented using XGBoost with the `multi:softprob` objective, allowing the model to estimate class probabilities for all three trading actions simultaneously.

During inference, the same feature construction pipeline is applied to the incoming financial context. The generated feature vector is passed to the trained meta-classifier, which predicts one of the three trading actions together with the corresponding class probabilities. Compared with static rule-based aggregation, the proposed meta-classification strategy automatically learns how different reasoning agents contribute under diverse market conditions, eliminating the need for manually predefined ensemble weights.

Table 1
Meta-Level Feature Representation

Feature Group	Features
Agent Actions	News, FinGPT, Event, Technical, Quantitative
Agent Confidence	News, FinGPT, Event, Technical, Quantitative
Technical Features	RSI, MACD
Market Features	Volatility, Trend Strength, Bollinger Band Width, Regime

6.2. Training Label Generation

Training labels for supervised learning were generated from future price movements computed over three prediction horizons (1-day, 3-day, and 5-day). For each historical observation, the future return was computed as

$$R_h = \frac{P_{t+h} - P_t}{P_t}, \tag{5}$$

where P_t denotes the current closing price and P_{t+h} represents the closing price after a forecasting horizon of h trading days.

Instead of using a fixed decision threshold, the proposed framework employs a volatility-adjusted threshold defined as

$$\theta = 0.8 \times \sigma, \tag{6}$$

where σ denotes the historical market volatility estimated during feature engineering.

The final trading labels are assigned according to

$$y = \begin{cases} \text{BUY}, & R_h > \theta \\ \text{SELL}, & R_h < -\theta \\ \text{HOLD}, & \text{otherwise.} \end{cases} \tag{7}$$

This volatility-aware labeling strategy adapts the decision boundaries to changing market conditions, producing more stable supervisory signals than fixed return thresholds. Independent target labels are generated for the 1-day, 3-day, and 5-day forecasting horizons and are subsequently used to train the corresponding XGBoost classifiers.

6.3. Decision Confidence Estimation

The confidence associated with the final recommendation is obtained directly from the probability distribution estimated by the XGBoost meta-classifier [7]. Specifically, the confidence score is defined as the maximum probability among the three possible trading actions:

$$Confidence = \max\left(P(BUY), P(HOLD), P(SELL)\right) \tag{8}$$

This confidence measure corresponds to the posterior probability assigned to the predicted class and is widely adopted in probabilistic classification models because it provides a simple and computationally efficient estimate of prediction certainty, making it suitable for real-time financial decision making [31, 32].

where $P(\cdot)$ denotes the probability assigned by the meta-classifier to each trading action. A higher confidence value indicates stronger agreement between the learned decision boundaries and the current financial context, while lower confidence values indicate greater predictive uncertainty in the final recommendation [31].

The specialized reasoning agents also provide confidence information that is incorporated into the meta-level feature vector. For learning-based agents, including the News Sentiment Agent, Financial Semantic Agent (FinGPT), and Multi-Horizon Quantitative Agent, confidence is computed as the maximum predicted class probability returned by the underlying model. For rule-based agents, including the Technical Analysis Agent and Event Detection Agent, confidence is represented by heuristic confidence scores derived from the strength and consistency of the corresponding financial signals. These values are treated as auxiliary input features for the meta-classifier rather than calibrated probability estimates [32, 33].

6.4. Safety Verification

To improve operational robustness during real-time deployment, the framework incorporates a lightweight safety verification stage after meta-classifier inference. Predictions associated with low confidence or unstable market conditions are conservatively converted into HOLD recommendations to reduce unnecessary trading risk. Additional safety constraints based on abnormal volatility and extreme technical indicator values further prevent unreliable predictions during highly uncertain market conditions. This post-processing stage improves decision stability while maintaining the low inference latency required by the FinMMEval Task 3 evaluation protocol.

6.5. Decision Rationale Generation

In addition to predicting the final trading action, the framework generates a concise natural-language rationale that summarizes the primary evidence supporting the recommendation. The explanation is produced using a lightweight template-based strategy that combines the outputs of the specialized reasoning agents with the final meta-classifier prediction. This design improves interpretability while avoiding unnecessary computational overhead during real-time inference.

The generated rationale highlights the most influential financial signals, including market sentiment, technical indicators, detected financial events, institutional filing information, and multi-horizon forecasting outcomes. To satisfy the FinMMEval Task 3 output requirements, the explanation is restricted to a maximum of 50 words.

An example rationale generated by the framework is shown below:

> *"Positive earnings sentiment, bullish momentum indicators, and improving medium-term forecasts suggest continued upward market movement. Technical trend strength and favorable institutional outlook support a BUY recommendation."*

7. Dataset Construction

The proposed framework was evaluated using the official CLEF 2026 FinMMEval Task 3 benchmark dataset [13, 14, 15]. The dataset provides daily JSON-based financial contexts for Bitcoin (BTC) and Tesla (TSLA), where each sample contains historical market prices, financial news articles, momentum annotations, future price differences, and optional SEC 10-K/10-Q corporate filings. The objective of the task is to predict a BUY, HOLD, or SELL recommendation together with a concise natural-language rationale for each daily financial context.

To support quantitative forecasting and feature engineering, additional historical market data were collected from Yahoo Finance [34]. The collected assets include AAPL, NVDA, MSFT, META, AMZN, SPY, QQQ, BTC, ETH, and SOL. These historical data were used to compute technical indicators, volatility measures, trend-related features, and other engineered market variables required by the Technical Analysis Agent and the Multi-Horizon Quantitative Agent.

Financial text processing was performed using the domain-specific FinBERT model [5]. Daily financial news articles were analyzed to estimate market sentiment and semantic financial signals, while optional SEC filings were incorporated to provide complementary long-term fundamental information. The resulting numerical and textual features constitute the heterogeneous financial inputs processed by the proposed multi-agent framework.

8. Feature Engineering

The feature engineering module transforms heterogeneous financial information into structured numerical representations that can be processed by the quantitative forecasting models and the meta-agent. Historical market prices obtained from the daily financial context are used to derive multiple technical indicators, volatility measurements, and trend-related features that summarize the current market state.

The extracted feature set includes Relative Strength Index (RSI), Moving Average Convergence Divergence (MACD), Simple Moving Averages (SMA), Exponential Moving Averages (EMA), Average True Range (ATR), Bollinger Bands, historical returns, rolling market volatility, and trend-strength indicators. These engineered variables provide complementary information regarding market momentum, price trends, volatility, and trading conditions, enabling the framework to capture both short-term fluctuations and medium-term market behaviour.

The Relative Strength Index (RSI) is computed as

$$RSI = 100 - \frac{100}{1 + RS}, \tag{9}$$

where

$$RS = \frac{\text{Average Gain}}{\text{Average Loss}}. \tag{10}$$

RSI measures the relative strength of recent upward and downward price movements and is used to identify potential overbought and oversold market conditions.

Market momentum is further characterized using the Moving Average Convergence Divergence (MACD), defined as

$$MACD = EMA_{12} - EMA_{26}, \tag{11}$$

where EMA_{12} and EMA_{26} denote the 12-period and 26-period exponential moving averages, respectively. Positive MACD values indicate bullish momentum, whereas negative values suggest weakening market trends.

In addition, rolling volatility, Bollinger Band statistics, and ATR are incorporated to characterize market uncertainty and price dispersion. These volatility-sensitive features complement momentum-based indicators by capturing abnormal market movements that frequently occur during periods of elevated financial uncertainty.

The complete engineered feature vector is supplied to the Technical Analysis Agent, the Multi-Horizon Quantitative Agent, and the meta-classification model, where it is combined with the outputs of the specialized reasoning agents to generate the final BUY, HOLD, or SELL recommendation.

9. Experimental Setup

The proposed framework was implemented in Python using PyTorch [35], Hugging Face Transformers [36], XGBoost [7], Pandas [37], NumPy [38], TA-Lib, and FastAPI [27]. The complete inference pipeline was deployed using Docker containers on Hugging Face Spaces, enabling lightweight real-time financial decision support compatible with the FinMMEval Task 3 evaluation protocol.

Historical market prices were first processed through the feature engineering module to compute technical indicators, volatility measures, trend-related statistics, and other engineered financial features. The resulting numerical features, together with financial news, momentum annotations, and optional SEC filings, were provided as inputs to the specialized reasoning agents. Each reasoning agent independently generated a BUY, HOLD, or SELL prediction together with an associated confidence score. These outputs were subsequently combined by the supervised meta-classification model to produce the final trading recommendation.

The quantitative forecasting component consists of three independent XGBoost classification models corresponding to prediction horizons of one day, three days, and five days. The models were configured with a maximum tree depth of six, a learning rate of 0.05, and 300 decision trees. Unless otherwise specified, model development followed an 80%/20% train-test split for internal evaluation.

XGBoost was selected because it efficiently models nonlinear relationships among heterogeneous numerical features while maintaining low computational overhead during inference. Compared with more computationally intensive deep learning models, XGBoost provides strong predictive performance together with low-latency execution, making it well suited for real-time financial decision making under the operational constraints of the FinMMEval Task 3 benchmark.

Table 2
XGBoost Hyperparameter Configuration

Parameter	Multi-Horizon	Meta-Classifier
Objective	`multi:softprob`	`multi:softprob`
Maximum Tree Depth	7	6
Learning Rate	0.03	0.025
Number of Trees	300	500
Subsample Ratio	0.80	0.85
Column Sample Ratio	0.80	0.85
Train-Test Split	80% / 20%	80% / 20% (Stratified)
Random Seed	Default	42
Output Classes	BUY/HOLD/SELL	BUY/HOLD/SELL

The multi-horizon forecasting models were trained using an 80%/20% train-test split with stratified sampling and a fixed random seed of 42 to ensure reproducible experiments.

System performance was evaluated using the official FinMMEval Task 3 financial evaluation metrics, including Cumulative Return (CR), Sharpe Ratio (SR), Maximum Drawdown (MD), Daily Volatility (DV), and Annualized Volatility (AV). Internal backtesting was performed by simulating sequential daily trading decisions using the predicted BUY, HOLD, and SELL actions. Final shared-task evaluation was conducted independently through the official FinMMEval evaluation platform.

9.1. Implementation Details

To support real-time deployment, the inference pipeline was optimized for low-latency prediction. The FastAPI service receives a daily financial context in JSON format, performs feature engineering and multi-agent reasoning, executes meta-agent inference, applies safety verification, and returns the final BUY, HOLD, or SELL recommendation together with its confidence score and a concise natural-language rationale. This modular deployment architecture enables efficient and reproducible financial decision making while satisfying the execution constraints of the FinMMEval Task 3 benchmark.

9.2. Computational Complexity Analysis

The proposed framework consists of multiple specialized reasoning agents operating sequentially on a daily financial context. Let N denote the number of input news articles, T the number of historical price observations, L the average token length of a financial document, and d the dimensionality of the engineered feature vector.

The computational complexity of the major components is summarized below:

Table 3
Computational Complexity of Major Components

Component	Complexity
News Sentiment Agent (FinBERT)	$O(NL)$
Technical Analysis Agent	$O(T)$
Event Detection Agent	$O(N)$
Filing Analysis Agent	$O(L)$
Financial Semantic Agent	$O(L)$
Multi-Horizon Forecasting	$O(T)$
Feature Fusion	$O(d)$
XGBoost Meta-Classifier	$O(MD)$
Safety Verification	$O(1)$

Here, M denotes the number of decision trees in the trained XGBoost model and D represents the maximum tree depth. Since the meta-classifier performs inference using a fixed number of trees, its prediction latency remains efficient for real-time financial decision making. The overall computational complexity of the proposed framework is dominated by transformer-based financial text processing and XGBoost inference, making the system suitable for low-latency deployment under the operational constraints of the FinMMEval Task 3 benchmark.

10. Experimental Results

10.1. Multi-Horizon Forecasting Results

The Multi-Horizon Quantitative Agent combines the probabilistic outputs of three independent forecasting models corresponding to 1-day, 3-day, and 5-day prediction horizons. An example inference generated by the framework produced prediction probabilities of 0.4662 for BUY, 0.2862 for SELL, and 0.2476 for HOLD, resulting in a final BUY recommendation. By integrating multiple forecasting horizons, the framework reduces the influence of short-term market fluctuations while preserving medium-term directional market information. This probabilistic fusion contributes to more stable trading decisions than relying on a single prediction horizon.

10.2. Backtesting Results

The proposed framework was further evaluated through internal backtesting using historical financial data under a simulated daily trading environment that follows the FinMMEval Task 3 reasoning-to-action protocol. Beginning with an initial simulated capital of $100,000, the system executed sequential BUY, HOLD, and SELL decisions generated by the complete multi-agent framework.

The framework executed a total of 468 simulated trades, including 211 BUY, 174 SELL, and 83 HOLD recommendations. The relatively large number of HOLD decisions demonstrates that the system does not force trades under uncertain market conditions and instead adopts conservative decisions when prediction confidence is insufficient. This behaviour contributes to limiting unnecessary market exposure while maintaining competitive cumulative returns.

The proposed framework achieved a cumulative return of 208.74% while maintaining a maximum drawdown of only 5.2%, indicating favourable risk-adjusted performance. The Sharpe Ratio of 0.6999

Table 4
Internal Backtesting Performance

Metric	Value
Initial Simulated Capital	\$100,000
Total Trades	468
Win Rate	73.08%
Final Portfolio Value	\$308,746.25
Cumulative Return (CR)	+208.74%
Sharpe Ratio (SR)	0.6999
Maximum Drawdown (MD)	-5.2%

further suggests that the generated returns were obtained with relatively controlled portfolio volatility. These results demonstrate that combining heterogeneous reasoning agents through the proposed meta-agent architecture can produce stable financial decision-making across varying market conditions.

The reported results correspond to internal backtesting experiments and are presented to evaluate the behaviour of the proposed framework. Official FinMMEval Task 3 leaderboard scores are obtained independently through the shared-task evaluation platform.

10.3. Ablation Study

To evaluate the contribution of each reasoning component, an ablation study was performed by progressively enabling individual agents within the proposed framework while keeping the remaining experimental settings unchanged. Starting from a sentiment-only baseline, additional reasoning modules were incorporated sequentially to measure their impact on overall trading performance.

Table 5
Ablation Study Results

Model Configuration	CR	SR
Sentiment-Only Model	+61.42%	0.31
Sentiment + Technical Analysis	+104.83%	0.46
+ Event Detection Agent	+146.27%	0.57
+ Filing Analysis Agent	+171.94%	0.63
+ Multi-Horizon Forecasting	+193.51%	0.67
Full Meta Classification layer Framework	+208.74%	0.6999

The results demonstrate that each additional reasoning component consistently improves both cumulative return and risk-adjusted performance. Adding the Technical Analysis Agent substantially improves market trend recognition beyond sentiment-only reasoning, while the Event Detection Agent further enhances performance by incorporating market-moving financial events that are not fully captured through sentiment analysis alone.

The Filing Analysis Agent provides complementary long-term fundamental information from corporate regulatory filings, resulting in additional performance gains. Incorporating the Multi-Horizon Quantitative Agent further improves temporal consistency by combining short-term and medium-term market forecasts, thereby reducing sensitivity to short-term market fluctuations.

The complete framework achieves the best overall performance, obtaining a cumulative return of 208.74% and a Sharpe Ratio of 0.6999. These results indicate that the specialized reasoning agents provide complementary information and that their integration through the proposed meta-agent produces more robust financial decision making than any individual reasoning component alone.

10.4. Baseline Comparison

The proposed framework was compared with several representative financial forecasting approaches, including sentiment-based prediction, machine learning, deep learning, transformer-based forecasting, and a conventional static ensemble model. All models were evaluated under the same internal backtesting protocol using cumulative return (CR) and Sharpe Ratio (SR) as the primary performance metrics.

Table 6
Baseline Comparison Results

Model	CR	SR
FinBERT Sentiment Model	+61.42%	0.31
XGBoost Forecasting Model	+97.18%	0.44
LSTM-Based Forecasting	+118.63%	0.49
Transformer Forecasting Model	+134.85%	0.53
Static Ensemble Model	+169.12%	0.61
Proposed Multi-Agent Framework	+208.74%	0.6999

The proposed framework consistently outperformed all baseline models in both cumulative return and risk-adjusted performance. Compared with sentiment-only prediction, the multi-agent architecture achieved substantially higher cumulative returns by integrating complementary reasoning from financial news, technical indicators, event detection, filing analysis, and quantitative forecasting. Compared with deep learning and transformer-based forecasting models, the proposed framework produced more stable trading performance by combining heterogeneous financial information instead of relying on a single predictive model. Furthermore, the improvement over the static ensemble demonstrates the effectiveness of the supervised meta-agent in integrating complementary reasoning signals into a unified trading decision.

10.5. Results Discussion

The experimental results demonstrate that combining heterogeneous financial reasoning agents provides more reliable trading decisions than relying on any individual information source. Sentiment analysis effectively captures market perception reflected in financial news, whereas technical indicators contribute robust trend and momentum information derived from historical prices. Event detection further improves responsiveness to sudden market-moving developments, while filing analysis provides complementary long-term fundamental information.

The Multi-Horizon Quantitative Agent enhances temporal consistency by combining forecasts across multiple prediction horizons, thereby reducing sensitivity to short-term market fluctuations. The supervised meta-agent integrates the outputs of all specialized reasoning agents together with engineered market features, enabling the framework to exploit complementary information from both textual and numerical financial data.

The internal backtesting and ablation results indicate that each reasoning component contributes positively to overall system performance. In particular, the progressive improvement observed during the ablation study suggests that the specialized reasoning agents capture different aspects of financial market behaviour, and their combination leads to more robust and stable trading decisions across diverse market conditions.

11. Real-Time Deployment

To support practical financial decision making, the proposed framework was deployed as a lightweight inference service using FastAPI, Docker, and Hugging Face Spaces. The deployment architecture enables real-time processing of daily financial contexts while satisfying the execution constraints of the FinMMEval Task 3 evaluation protocol.

During inference, the FastAPI service receives a JSON-formatted financial context containing historical market prices, financial news articles, momentum annotations, historical price trajectories, and optional SEC 10-K/10-Q corporate filings. The incoming data are first preprocessed and transformed into engineered numerical features before being processed by the specialized reasoning agents.

Each reasoning agent independently generates a BUY, HOLD, or SELL recommendation together with an associated confidence score. These outputs are combined by the trained meta-classification model, which produces the final trading decision and its corresponding prediction confidence. A lightweight safety verification stage is subsequently applied to reduce unreliable predictions under highly uncertain market conditions. Finally, the system returns a structured JSON response containing the recommended trading action, confidence score, and a concise natural-language rationale.

To improve operational robustness, the deployment pipeline incorporates a fail-safe mechanism that automatically returns a HOLD recommendation whenever inference cannot be completed successfully because of unexpected execution failures or deployment constraints. This conservative strategy prevents unintended trading actions and ensures stable behaviour during real-time financial decision support.

12. Limitations

Although the proposed framework demonstrates encouraging performance on the FinMMEval Task 3 benchmark and internal backtesting experiments, several limitations remain. First, the reported trading performance is based on historical market simulation and may not fully reflect real-world trading environments, where transaction costs, slippage, liquidity constraints, execution latency, and market impact can significantly influence realized returns.

Second, the quality of the generated recommendations depends on the availability and reliability of heterogeneous financial information, including financial news, historical market prices, momentum annotations, and optional corporate filings. Missing, delayed, or noisy financial data may reduce the effectiveness of the specialized reasoning agents and consequently affect the final trading decision.

Third, although the proposed multi-agent framework improves decision robustness by integrating complementary reasoning sources, the supervised meta-classification model is trained using historical market data and may require periodic retraining to maintain performance under evolving market conditions. Financial markets are inherently non-stationary, and relationships learned from historical data may not generalize perfectly to future market behaviour.

Finally, the current framework employs lightweight rule-based reasoning for technical analysis and event detection to satisfy the real-time execution constraints of the FinMMEval benchmark. Future work will investigate more sophisticated reasoning strategies, adaptive online learning techniques, and larger financial language models while preserving the low-latency requirements of practical financial decision support.

13. Error Analysis

To better understand the behavior of the proposed framework, we analyze several representative situations in which the system may generate suboptimal trading decisions. The observed errors primarily arise from the complexity and uncertainty of financial markets rather than failures of individual reasoning agents.

1. **Conflicting reasoning signals.** In some market situations, different reasoning agents produce contradictory recommendations. For example, financial news may indicate positive market sentiment while technical indicators suggest a bearish trend. Although the meta-classification model learns to combine these heterogeneous signals, conflicting evidence may occasionally reduce prediction confidence and result in conservative HOLD recommendations.
2. **Unexpected market events.** Sudden macroeconomic announcements, geopolitical developments, or unforeseen financial events may cause abrupt market movements that cannot be fully anticipated using historical market data or previously observed financial patterns.

3. **Short-term market noise.** Rapid price fluctuations during sideways or highly volatile markets can temporarily affect the Multi-Horizon Quantitative Agent, particularly for shorter prediction horizons. Although combining multiple forecasting horizons improves temporal stability, isolated short-term fluctuations may still influence individual predictions.
4. **Incomplete financial information.** The performance of the framework depends on the availability and quality of financial news articles, historical market prices, momentum annotations, and optional SEC filings. Missing or delayed information may reduce the effectiveness of the specialized reasoning agents.
5. **Rule-based reasoning limitations.** The Technical Analysis Agent and Event Detection Agent employ lightweight rule-based reasoning to satisfy real-time inference requirements. While computationally efficient, these deterministic rules may not capture every complex market scenario that could potentially be modeled by more sophisticated learning-based approaches.
6. **Real-time deployment constraints.** The framework is optimized for low-latency inference under the execution constraints of the FinMMEval Task 3 benchmark. Consequently, computationally intensive reasoning strategies are intentionally avoided, potentially limiting performance in situations requiring deeper financial analysis.

14. Conclusion and Future Work

This paper presented a Multi-Agent Financial Intelligence Framework for reasoning-to-action financial decision making developed for the CLEF 2026 FinMMEval Task 3 benchmark. The proposed framework integrates heterogeneous financial information, including financial news, historical market prices, technical indicators, corporate filings, financial events, and multi-horizon quantitative forecasting within a unified decision-making pipeline. Specialized reasoning agents independently analyze complementary aspects of the financial environment, while a supervised meta-classification model combines their outputs to generate the final BUY, HOLD, or SELL recommendation together with an associated confidence score.

Experimental evaluation through internal backtesting demonstrated that the proposed framework consistently outperformed individual forecasting models and conventional static ensemble approaches in terms of cumulative return and risk-adjusted performance. The ablation study further showed that each reasoning component contributes positively to the overall system, highlighting the benefit of integrating heterogeneous financial information for practical financial decision support.

Future work will focus on improving the robustness and generalization capability of the framework through larger financial language models, retrieval-augmented financial reasoning,feature learning, and multimodal financial intelligence. We also plan to investigate online model updating, more advanced event understanding, and broader evaluation across additional financial assets and market conditions while preserving the low-latency requirements of real-time financial decision support.

Declaration on Generative AI

Generative AI tools were used during the preparation of this manuscript solely to assist with language refinement, grammar correction, and improving the clarity and organization of the presentation. Generative AI was also used to support literature exploration, brainstorming of writing ideas, and code documentation during system development. All methodological decisions, system design, implementation, experimental evaluation, and scientific conclusions were developed, verified, and validated by the authors. The authors carefully reviewed and edited all AI-assisted content and take full responsibility for the accuracy, originality, and integrity of the final manuscript.

References

[1] E. F. Fama, Efficient capital markets: A review of theory and empirical work, The Journal of Finance 25 (1970) 383–417.

[2] A. W. Lo, The adaptive markets hypothesis, The Journal of Portfolio Management 30 (2004) 15–29.

[3] X. Ding, Y. Zhang, T. Liu, J. Duan, Deep learning for event-driven stock prediction, in: Proceedings of IJCAI, 2015, pp. 2327–2333.

[4] Z. Hu, et al., Event-driven financial forecasting: A survey, ACM Computing Surveys (2020).

[5] D. Araci, Finbert: Financial sentiment analysis with pre-trained language models, arXiv preprint arXiv:1908.10063 (2019).

[6] H. Y. et al., Fingpt: Open-source financial large language models, arXiv preprint arXiv:2306.06031 (2023).

[7] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of KDD, 2016.

[8] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Computation 9 (1997) 1735–1780.

[9] A. Vaswani, et al., Attention is all you need, in: Advances in Neural Information Processing Systems, 2017.

[10] B. Lim, S. O. Arik, N. Loeff, T. Pfister, Temporal fusion transformers for interpretable multi-horizon time series forecasting, International Journal of Forecasting 37 (2021) 1748–1764.

[11] Y. Nie, N. H. Nguyen, P. Sinthong, J. Kalagnanam, A time series is worth 64 words: Long-term forecasting with transformers, International Conference on Learning Representations (2023).

[12] Y. Zhang, et al., A survey of deep learning for financial time series forecasting, Artificial Intelligence Review (2023).

[13] Z. Xie, Y. Dai, R. Elbadry, V. Jani, X. Peng, L. Qian, G. Georgiev, D. Dimitrov, F. Zhang, J. Huang, J. Geng, Y. Chen, Y. Yuan, H. Wu, Y. Wang, I. Koychev, V. Stoyanov, M. Song, Y. Chen, X. Liu, P. Nakov, Overview of FinMMEval 2026: Multilingual and multimodal financial evaluation, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction, Proceedings of the Seventeenth International Conference of the CLEF Association (CLEF 2026), Springer Lecture Notes in Computer Science LNCS, Jena, Germany, 2026.

[14] Z. Xie, L. Qian, G. Georgiev, D. Dimitrov, R. Elbadry, F. Zhang, X. Peng, J. Huang, V. Jani, Y. Dai, J. Geng, Y. Chen, Y. Yuan, H. Wu, Y. Wang, I. Koychev, V. Stoyanov, M. Song, Y. Chen, X. Liu, P. Nakov, Overview of the FinMMEval 2026 task 3: Financial decision making, in: CLEF 2026 Working Notes, CEUR Workshop Proceedings, CEUR-WS.org, Jena, Germany, 2026.

[15] L. Qian, X. Peng, Y. Wang, V. J. Zhang, H. He, H. Smith, Y. Han, Y. He, H. Li, Y. Cao, Y. Yu, A. Lopez-Lira, P. Lu, J.-Y. Nie, G. Xiong, J. Huang, S. Ananiadou, When agents trade: Live multi-market trading benchmark for LLM agents, 2025. URL: https://arxiv.org/abs/2510.11695. doi:10.48550/arXiv.2510.11695. arXiv:2510.11695.

[16] D. H. Wolpert, Stacked generalization, Neural Networks 5 (1992) 241–259.

[17] T. G. Dietterich, Ensemble methods in machine learning, Multiple Classifier Systems (2000) 1–15.

[18] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of NAACL-HLT, 2019.

[19] S. Wu, et al., Bloomberggpt: A large language model for finance, arXiv preprint arXiv:2303.17564 (2023).

[20] L. Breiman, Random forests, Machine Learning 45 (2001) 5–32.

[21] G. Ke, et al., Lightgbm: A highly efficient gradient boosting decision tree, in: Advances in Neural Information Processing Systems, 2017.

[22] K. Cho, et al., Learning phrase representations using rnn encoder–decoder for statistical machine translation, arXiv preprint arXiv:1406.1078 (2014).

[23] D. Stempień, R. Ślepaczuk, Hybrid models for financial forecasting: combining econometric, machine learning, and deep learning models, arXiv preprint arXiv:2505.19617 (2025).

[24] M. P. Clements, P. H. Franses, N. R. Swanson, Forecasting economic and financial time-series with non-linear models, International journal of forecasting 20 (2004) 169–183.

[25] L. Breiman, Bagging predictors, Machine Learning 24 (1996) 123–140.

[26] L. Rokach, Ensemble-based classifiers, Artificial Intelligence Review 33 (2010) 1–39.
[27] S. Ramírez, Fastapi documentation, https://fastapi.tiangolo.com/, 2024.
[28] J. W. Wilder, New Concepts in Technical Trading Systems, Trend Research, 1978.
[29] J. J. Murphy, Technical Analysis of the Financial Markets, New York Institute of Finance, 1999.
[30] J. Bollinger, Bollinger on Bollinger Bands, McGraw-Hill, 2001.
[31] C. Guo, G. Pleiss, Y. Sun, K. Q. Weinberger, On calibration of modern neural networks, in: Proceedings of the 34th International Conference on Machine Learning (ICML), 2017, pp. 1321–1330.
[32] J. C. Platt, Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, Advances in Large Margin Classifiers (1999) 61–74.
[33] B. Zadrozny, C. Elkan, Transforming classifier scores into accurate multiclass probability estimates, in: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002, pp. 694–699.
[34] R. Aroussi, yfinance: Yahoo finance market data downloader, https://github.com/ranaroussi/yfinance, 2024.
[35] A. P. et al., Pytorch: An imperative style, high-performance deep learning library, in: Advances in Neural Information Processing Systems (NeurIPS), 2019.
[36] T. W. et al., Transformers: State-of-the-art natural language processing, in: Proceedings of EMNLP, 2020.
[37] W. McKinney, Data structures for statistical computing in python, in: Proceedings of the 9th Python in Science Conference, 2010.
[38] C. R. H. et al., Array programming with numpy, Nature (2020).