International Journal of Science, Strategic Management and Technology
Volume **02** Issue **05** May-**2026** | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

AI-Driven Trading Analysis Platform:Architecture, Models, Risk Controls, and Operational Framework

Ruban S, Raghu Nandha Kumar D.E, Sithik Raj R
School of Computing Sciences, VISTAS, Pallavaram, Chennai, India

https://doi.org/10.55041/ijsmt.v2i5.014

Cite this Article: S, R., D.E, R. N. K. & R, S. R. (2026). AI-Driven Trading Analysis Platform:Architecture, Models, Risk Controls, and Operational Framework. International Journal of Science, Strategic Management and Technology, *02*(05). https://doi.org/10.55041/ijsmt.v2i5.014

Abstract—The Trading AI Analysis platform is an intelligent, modular system designed to support systematic trading decisions through the integration of machine learning, real-time market data pipelines, and quantitative signal generation. Built on an event-driven architecture comprising five principal layers — Data Ingestion, Feature Engineering, Model Layer, Signal Engine, and Risk & Execution — the platform processes real-time and historical market data. The system deploys an ensemble of AI models including LSTM networks, Transformer architectures, Gradient Boosting, and XGBoost classifiers. A structured signal schema delivers directional recommendations with confidence scores and risk flags. Robust pre-trade and real-time risk controls enforce position limits, circuit breakers, and compliance constraints. All signals and decisions are persisted in an immutable audit log ensuring full regulatory compliance over a seven-year retention period.

Keywords—algorithmic trading, machine learning, LSTM, XGBoost, transformer, signal generation, risk management, quantitative finance, real-time systems, regulatory compliance.

I. INTRODUCTION

The proliferation of algorithmic and high-frequency trading has fundamentally transformed modern financial markets. Systematic trading strategies now account for a substantial proportion of daily equity and derivatives volume, necessitating robust, low-latency infrastructure that can process and act on market signals within milliseconds.

Modern financial markets demand not only speed but also intelligent adaptability to changing market conditions. AI-driven platforms enable predictive analytics, automated execution, and risk-aware decision-making, thereby reducing human bias and operational inefficiencies. Furthermore, the integration of big data technologies allows systems to process diverse data sources such as news sentiment, macroeconomic indicators, and social media signals.

This paper presents the Trading AI Analysis (TAI) platform, a production-grade system that integrates machine learning model ensembles, real-time data pipelines, and quantitative signal generation to support fully automated trading decisions. The platform is designed to serve quantitative analysts, software engineers, risk managers, and compliance officers working within algorithmic trading infrastructure.

The remainder of this paper is organized as follows: Section II describes the system architecture; Section III

details the AI models and feature engineering; Section IV covers risk controls; Section V presents the API and integration layer; Section VI describes the backtesting framework; Section VII addresses monitoring and alerting; Section VIII discusses compliance and auditability; and Section IX concludes with a glossary.

II. SYSTEM ARCHITECTURE

The TAI platform is built as a modular, event-driven architecture comprising five principal layers, each with clearly delineated responsibilities and technology stacks. This separation of concerns enables independent scaling, fault isolation, and continuous deployment of individual components.

A. Architectural Layers

The five layers are described in Table I.

TABLE I. SYSTEM ARCHITECTURE LAYERS

Layer	Description	Technology
Data Ingestion	Real-time and historical market data from exchanges, alternative data vendors, and news feeds.	Kafka, Redis, TimescaleDB
Feature Engineering	Computation of technical indicators, statistical features, order-book microstructure, and sentiment signals.	Python, pandas, TA-Lib
Model Layer	Ensemble of ML models including gradient boosting, LSTM, and transformer-based architectures.	PyTorch, XGBoost, scikit-learn
Signal Engine	Combines raw model outputs into actionable signals with confidence scores and position-size recommendations.	Custom C++ / Python bridge
Risk & Execution	Pre-trade risk checks, portfolio-level constraints, and order routing to execution venues.	FIX 4.4, internal risk engine

B. Data Flow

Data enters the system via the Ingestion Layer, where it is normalized and stored in TimescaleDB, then streamed to the Feature Engineering service. Computed features are published to a Kafka topic, consumed by the Model Layer in real time. The Signal Engine aggregates model votes and emits structured signal objects, which are evaluated by the Risk & Execution layer before reaching the order management system (OMS).

III. AI MODELS AND FEATURE ENGINEERING

A. Model Inventory

The platform deploys a four-model ensemble, each targeting a distinct prediction objective. Table II summarizes the model inventory.

TABLE II. AI MODEL INVENTORY

Model ID	Type	Prediction Target	Lookback	Update Freq.
TAI-M01	LSTM (2-layer)	5-min return direction	60 bars	Hourly
TAI-M02	XGBoost	Volatility regime	120 bars	Daily
TAI-M03	Transformer	Sentiment + price fusion	240 bars	Daily

Model ID	Type	Prediction Target	Lookback	Update Freq.
TAI-M04	Gradient Boost	Order-flow imbalance	30 bars	Real-time

B. Feature Engineering

The feature set spans four categories, providing the model ensemble with a comprehensive view of market conditions.

Price & Volume Features:

- OHLCV bars at 1-min, 5-min, 15-min, 1-hr, and 1-day granularity
- Rolling returns over 1, 5, 10, 20, and 60-bar windows
- VWAP and deviation from VWAP
- Relative volume vs. 20-day rolling average

Technical Indicators:

- RSI (14-period), MACD (12/26/9), Bollinger Bands (20/2)
- ATR (14), ADX (14), CCI (20)
- Moving averages: SMA-20, SMA-50, EMA-9, EMA-21

Market Microstructure:

- Bid-ask spread, mid-price, and top-of-book depth
- Order flow imbalance (OFI) at L1 and L2
- Trade-initiated buy/sell ratio

Alternative Data:

- News sentiment score (NLP model, range: -1.0 to +1.0)
- Social media mention velocity (normalized)
- Earnings surprise factor (for equities)

C. Signal Output Schema

Each signal published by the Signal Engine conforms to a structured JSON schema. Key fields include: symbol (asset identifier), timestamp (UTC milliseconds), direction (BUY | SELL | HOLD), confidence (float [0.0, 1.0]), signal_strength (float [-1.0, 1.0]), suggested_size (% of portfolio), model_votes (per-model breakdown), and risk_flags (active risk conditions such as HIGH_VOLATILITY or NEWS_EVENT).

IV. RISK CONTROLS

The Risk & Execution layer enforces a layered set of pre-trade and real-time risk controls before any signal reaches the order management system. This multi-tiered approach ensures that individual signal risk is evaluated both in isolation and within the broader portfolio context.

A. Pre-Trade Checks

- Maximum single-order notional value: configurable per instrument class
- Position limit enforcement: long and short caps per symbol, sector, and portfolio
- Concentration check: maximum percentage of ADV per order
- Drawdown circuit breaker: halts signal generation if intraday portfolio drawdown exceeds threshold
- News blackout window: signals suppressed N minutes before/after scheduled news events

B. Risk Parameters

Default risk parameters are defined in Table III and are configurable at the per-instrument, strategy, or risk-manager level.

TABLE III. DEFAULT RISK PARAMETERS

Parameter	Default Value	Override Level
Max notional per order	$500,000	Per-instrument config
Max portfolio position	5% of NAV	Strategy config
Daily loss limit	2% of NAV	Risk manager override
Max ADV participation	10%	Per-instrument config
News blackout window	15 min pre / 5 min post	Per-symbol config
Min confidence threshold	0.62	Strategy config

V. API AND INTEGRATION LAYER

A. Authentication

All API requests require a Bearer token obtained from the IAM service. Tokens expire after 60 minutes and must be refreshed using the /auth/refresh endpoint. The base URL for all REST endpoints is: https://api.trading-ai.internal/v2.

B. Core REST Endpoints

TABLE IV. CORE API ENDPOINTS

Method	Endpoint	Description
GET	/signals/{symbol}	Latest signal for a symbol
GET	/signals/batch	Signals for up to 100 symbols
GET	/models/{model_id}/status	Model health & drift metrics
GET	/features/{symbol}	Current feature vector
POST	/backtest/run	Submit a backtest job
GET	/backtest/{job_id}/results	Retrieve backtest results
GET	/risk/positions	Live position & risk exposure
POST	/risk/override	Submit a risk parameter override

C. WebSocket Streaming

For low-latency signal consumption, the platform provides a WebSocket endpoint at wss://stream.trading-ai.internal/v2/signals. Clients subscribe by sending a JSON message specifying target symbols and a minimum confidence threshold filter. This enables consumers to receive signals within single-digit milliseconds of generation.

VI. BACKTESTING FRAMEWORK

A. Configuration

Backtest jobs are submitted as JSON configuration objects via POST /backtest/run. Parameters include: start_date / end_date (ISO-8601), universe (symbol list or predefined set such as SP500 or NASDAQ100), strategy_config (signal thresholds, position sizing, rebalance frequency), cost_model (commission rates and slippage model), and risk_config (risk parameters for the simulation).

B. Performance Metrics

The framework reports the metrics listed in Table V, providing a comprehensive view of strategy risk-adjusted performance.

TABLE V. BACKTESTING PERFORMANCE METRICS

Metric	Definition
Total Return	Cumulative portfolio return over the backtest period
Sharpe Ratio	Annualized risk-adjusted return (excess return / annualized vol)
Sortino Ratio	Sharpe variant using downside deviation in denominator
Max Drawdown	Largest peak-to-trough decline in portfolio value
Calmar Ratio	Annualized return divided by maximum drawdown
Hit Rate	Percentage of trades that were profitable
Profit Factor	Gross profit divided by gross loss
Turnover	Annualized average portfolio turnover

VII. MONITORING AND ALERTING

A. Model Health Metrics

The platform continuously monitors model quality and data pipeline health via the /models/{model_id}/status endpoint. Tracked metrics include: prediction drift (KL

divergence between live and training distributions), feature drift (Population Stability Index, PSI), inference latency percentiles (p50/p95/p99), data freshness (seconds since last ingestion per source), and signal generation rate per symbol per hour.

B. Alert Thresholds

Table VI defines warning and critical thresholds for each monitored metric, along with the automated remediation action taken upon breach.

TABLE VI. MONITORING ALERT THRESHOLDS

Alert	Warning	Critical	Action
Prediction drift (KL div)	> 0.05	> 0.15	Auto-retrain trigger
Feature PSI	> 0.10	> 0.25	Page on-call quant
Inference latency p99	> 50ms	> 200ms	Scale-up autoscaler
Data freshness	> 30 sec	> 120 sec	Halt affected signals
Daily drawdown	> 1% NAV	> 2% NAV	Circuit breaker trip

VIII. COMPLIANCE AND AUDITABILITY

All signals, model predictions, feature snapshots, and executed orders are persisted in an immutable audit log stored in an append-only S3-compatible object store. The retention period is seven years, in accordance with applicable regulatory requirements.

- Every signal is tagged with the model version, feature hash, and operator identity that triggered it.
- Risk overrides require dual authorization and are logged with justification text.
- Model retraining events generate a training report archived alongside the model artifact.
- Monthly model performance reports are automatically generated and routed to risk management.

IX. CONCLUSION

This paper has presented the Trading AI Analysis platform, a modular, event-driven system that integrates machine learning model ensembles, real-time market data pipelines, structured signal generation, and multi-layered risk controls into a cohesive algorithmic trading infrastructure.

The platform's four-model ensemble — combining LSTM, XGBoost, Transformer, and Gradient Boosting architectures — enables robust prediction across multiple market dimensions including return direction, volatility regime, sentiment fusion, and order-flow dynamics. The structured signal schema, configurable risk controls, comprehensive backtesting framework, and immutable audit trail collectively provide a production-ready foundation for systematic, compliant, and auditable trading operations.

Future work will focus on expanding the alternative data ingestion pipeline, incorporating reinforcement learning-based execution optimization, and extending the platform's multi-asset capabilities to include fixed income and FX derivatives.

X. REFERENCES

[1] J. Sirignano and R. Cont, "Universal features of price formation in financial markets," Quantitative Finance, vol. 19, no. 9, pp. 1449-1459, 2019.

[2] B. Lim and S. Zohren, "Time-series forecasting with deep learning: a survey," Philosophical Transactions of the Royal Society A, vol. 379, 2021.

[3] A. Vaswani et al., "Attention is all you need," in Advances in Neural Information Processing Systems (NeurIPS), 2017.

[4] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. ACM SIGKDD, 2016, pp. 785-794.

[5] M. Lopez de Prado, Advances in Financial Machine Learning. Hoboken, NJ: Wiley, 2018.

[6] R. Kissell, Algorithmic Trading Methods, 2nd ed. Academic Press, 2020.

[7] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016.

[8] E. Chan, Algorithmic Trading: Winning Strategies and Their Rationale. Wiley, 2013.

[9] C. Bishop, Pattern Recognition and Machine Learning. Springer, 2006.

[10] A. Ng, Machine Learning Yearning, 2018.