

Abhishek R-P - AI stock trading strategy

What's a quick way to use an AI to trade, make trades for example? What I'm thinking about is like just the top of my head it's like subscribing to RSS feed of something like Economics Times etc. in India. It's for Indian stock market and then whenever a positive news launches as soon as it drops for a particular stock or a particular industry I want to use my AI agent to invest a certain amount. that I am willing to risk on that particular way so give me a whole structure, architecture of this what are the best possible ways like this is a very new thought in my mind but I'm thinking to explore more on this but yeah you give me every information everything you have plan strategy what is important what is not important how it has worked in the past for other people what works what doesn't work who all people are employing these things etc. where the risks are for me where the profit margins are for me how should I structure it that kind of deal you can think hard about this then give me a possible answer.

Great idea—and totally doable—but the winning version is **not** “read ET RSS → buy quickly.” By the time a headline hits mainstream RSS, the move is usually gone. In India, the *earliest, tradable* news is almost always the **exchange corporate announcement** (NSE/BSE) or a machine-readable vendor feed; mainstream portals are minutes late.

Below is a practical, low-BS blueprint you can actually build, plus what works, what doesn't, risks, and how others run this.

TL;DR: the winning shape

- **Signal source (fastest first):**
 1. **NSE/BSE corporate announcements** (official, timestamped) → strongest edge.

2. **Paid machine-readable feeds** (Reuters/Bloomberg/TickerPlant/etc.). Free or “cheap” APIs exist but are often delayed or EOD.
 3. **Mainstream RSS/Twitter** only as confirmation or for *slower* thematic bets.
- **Classifier before “AI”:** small, deterministic **event classifier** (buyback, order win, rating change, capex, pledge release, results) + a light sentiment model. LLMs are for *summaries/explanations*, not your trigger.
 - **Execution:** go through a regulated broker API (Zerodha Kite, Dhan, Angel, Upstox). Start with **discretionary order confirm**; later flip to full auto *after* approvals/logging are in place.
 - **Compliance (India, 2025):** SEBI now requires **tagged/approved algos via brokers**, provider registration, and auditability. Don’t skip this.
-

What actually works (and why)

Works

- **Exchange announcements → small/midcap drift.** Things like “large order win,” “buyback @ X,” “pledge release,” “rating upgrade,” “capacity expansion” often produce **tradeable 30–180 min moves** in less-covered names. (Largecaps react instantly; midcaps drift.)
- **Event rules > generic sentiment.** If *buyback price* is 10% above CMP and size \geq Y% of mcap → high-conviction long. If *order win value* \geq 1× quarterly revenue → positive drift. If *pledge increase* → often negative short-term.
- **Human-in-the-loop** in month 1–2. Let the system surface ranked, pre-sized trades; you click “Send.”

Usually doesn’t

- **“ET RSS → market”:** big desks and prop shops have already read the exchange PDF. You’ll chase.
 - **Pure LLM trading:** hallucinations, latency, and PDF tables confuse general models. Use rules + tiny domain model; use LLM only to *explain*.
-

Architecture (battle-tested layout)

1) Ingestion (multi-source, prioritized)

- **NSE/BSE Corporate Announcements:** scrape/poll the official pages with backoff, or (better) subscribe to a vendor with minute-level or real-time beats. Map ISIN \leftrightarrow symbol.
- **Tier-2:** Reuters/DJ machine-readable news if budget allows.
- **Tier-3:** Company X handles, broker blogs, ET/BS/Moneycontrol RSS \rightarrow confirm slower themes (sector momentum).

2) Normalization & entity linking

- Clean the text/PDF; extract fields (company, instrument, numbers, units, dates, rating agency, buyback price/size, order value, pledge %, etc.).
- Link **entity** \rightarrow **NSE/BSE symbol(s)** via ISIN; fall back to fuzzy on name + sector.

3) Event classifier (rules + micro-model)

- Rule signatures (fast, robust):
 - **Buyback:** “buy-back/buyback”, “maximum price ₹X”, “aggregate amount ₹Y crore”.
 - **Order win:** “received order/work order/LoA worth ₹X from ...”.
 - **Rating:** “CRISIL/ICRA/CARE/India Ratings upgrades/downgrades”.
 - **Pledge:** “promoter pledge [increase/decrease]”.
 - **Capex/expansion:** “greenfield/brownfield”, capex ₹X.
 - **Results:** YoY/ QoQ deltas.
- Sentiment/impact score: small fin-tuned classifier (distil-size) that outputs **impact** $\in [-1, +1]$ and **confidence**. Keep it deterministic where money is at stake.

4) Strategy engine

- **Event** \rightarrow **playbook** mapping, e.g.
 - *Buyback above CMP & size big* \rightarrow **CNC buy + 1–3 day hold** (or **call spread** on liquid names).
 - *Big order win vs revenue* \rightarrow **intraday to multi-day long** with trailing stop.
 - *Rating downgrade/pledge increase* \rightarrow **intraday short/futures** if liquid, otherwise avoid.
- Position sizing: **Kelly-lite** (e.g., risk 0.25–0.5% of equity/trade), max sector exposure caps, max 3 concurrent event bets.

5) Execution layer (broker APIs)

- OMS microservice (FastAPI) that creates **bracket/CO or GTT** with pre-set SL/TP, tags orders with **algo ID**, captures **order/ack timestamps**, and logs everything for audit.

- Start with “**confirm to execute**” via Telegram/Slack; swap to fully automatic only after your broker/exchange approvals.

6) Storage, audit, and research

- **Event store** (Postgres): announcement_id, time_seen, parsed fields, impact score.
- **Market store** (Parquet): 1-sec or tick bars for the symbol (you already have this).
- **Trade store**: end-to-end latencies, slippage, P&L, reason codes (“BUYBACK_>CMP10”).
- **Dashboards**: win rate by event type, average drift, liquidity filters that work, slippage by time-of-day.

7) Ops

- **Kill-switch**, risk guardrails (max daily loss, per-trade loss, max exposure), and circuit-breaker logic (e.g., disable after 3 consecutive fails).
- **Latency SLOs**: ingest→parse→decide→order in $\leq 2-5$ s for equity; options are tighter due to spreads.

Compliance & reality check (India, 2025)

- **SEBI/Exchange** now requires **tracking, tagging, and approvals** for retail algos offered via brokers; brokers must obtain exchange permission for each strategy, and orders must carry a **unique identifier**. Design your stack for **auditability** (full logs, replay, versioned strategy).
- SEBI is also proposing to bake algo trading oversight into the **stock-broker regulations**; expect more process, not less.
- Bottom line: **build it right** (logs, versioning, order tags) and clear it with your broker before flipping to unattended auto.

MVP you could ship in ~1–2 weekends

1. Data

- Poll NSE/BSE announcements (HTML/JSON endpoints) every 2–5 s with ETag/backoff; cache seen ids; save raw + parsed.
- (Optional) Add a paid corporate-announcement feed if budget allows (strongly preferred for reliability).

2. Parser + rules

- Regex/keyword rules for buyback/order win/rating/pledge/capex/results; compute *impact features* (e.g., buyback premium %, order_value/revenue, rating notch).

3. Scorer

- Light classifier: logistic/GBM on impact features; add a tiny transformer only to handle tricky language.

4. Strategy

- Start with **two high-edge plays**: (A) Buybacks with large premium/size; (B) Order wins $\geq 1\times$ quarterly revenue in small/midcaps.
- Risk: 0.3%/trade, fixed SL (ATR-based), time-based exit if no follow-through.

5. Execution

- Zerodha Kite Connect or your preferred broker API. Place bracket/GTT with tags; Telegram inline buttons: [Skip] [Half-Size] [Full].

6. Backtest

- Rebuild an **event study**: for each historical announcement, align to your market data at $T_0 = \text{time_seen} + \text{assumed latency (e.g., 5–30 s)}$; compute drift windows (5, 15, 60, 120 min; T+1 open).
- Track slippage, borrow costs (if short), and realistic lot sizes.

7. Governance

- Version every rule/model; store hashes in trade logs; nightly PDF of all day's trades/decisions for audit.

Edges, costs, and expectations

- **Edge**: This is *not* HFT; you're going for **information-driven drift** where the street is slow to quantify impact (esp. midcaps). Expect lots of *small* wins and quick scratches.
 - **Costs**: Feed + infra + API fees. Exchange-grade real-time announcement APIs can run into lakhs/year; free scraping risks rate limits and downtime.
 - **Where margins live**: clear, numeric events (buyback premium/size, order-value-to-revenue, rating notch). **Avoid** vague PR lines.
 - **Drawdowns**: headline reversals, mis-parses, option spread blowouts, and "sell-the-news" in crowded names.
-

Who uses this & how it's evolved

- **Prop/HFT and big funds** ingest machine-readable news and exchange PDFs in milliseconds. For retail, the competitive spot is **event drift on midcaps** and **well-structured corporate actions**.
 - India's regulator has tightened oversight in 2024–2025 (algo tagging/approvals) and is moving to formalize the framework. Plan for it from day one.
-

Quick build kit (stack suggestion)

- **Ingestion:** Python + httpx/Playwright; retry/backoff; Redis stream for fan-out.
 - **Parse/NER:** spaCy/Presidio + your rules; pdftotext or camelot for tables.
 - **Model:** LightGBM/XGBoost on engineered features; optional mini-transformer (distilbert-size) for confidence.
 - **Backtest:** DuckDB/Polars over your Parquet ticks; event-aligned windows.
 - **Execution:** FastAPI OMS + Zerodha/Dhan SDK; persistent order router; Telegram bot for confirm.
 - **Observability:** Prometheus + Grafana; alert on ingestion lag, parse failures, broker rejects.
 - **Governance:** per-order **algo_id**, config hash, data provenance, immutable logs.
-

Next steps I'd take (practical)

1. Pick **two event types** (buyback, order win).
2. Build the **announcement → trade** pipeline with manual confirm.
3. Backtest on your 1-sec data with a **+15 s latency assumption**; size conservatively.
4. Talk to your broker re: **algo tagging/approval** path; add audit logs now so you don't rework later.

Want me to sketch the actual rule set + feature schema (CSV columns) and a minimal FastAPI + broker-API scaffold you can paste into your repo? I can do that in one go.

Recent India algo-trading regulation moves

Reuters

**India markets regulator
sets track and trace
rules for retail investors'
algo trading**

Feb 4, 2025

Reuters

**India markets regulator
proposes to add algo
trading into stock
broker regulations**

5 days ago

Reuters

**How India struggled to
regulate Jane Street's
money-spinning
machine**

4 days ago

This is educational, not investment advice. Trading involves risk; comply with SEBI/broker rules before automating.

○○○ Sources