VT INTERNATIONAL PRESS
VTIPJMRR | Journal of Multidisciplinary Research and Review

AI-SMIS: An Explainable, Evidence-Grounded Framework for Stock Market Intelligence — System Architecture and Pre-Specified Evaluation Protocol

Vasudha Patil[1], Viraj Darandale[2], Siddesh Khandekar[3], Aditya Gawade[4]

[1]*Assistant Professor & HOD, Department of Artificial Intelligence and Data Science Engineering, Shri Chhatrapati Shivaji Maharaj College of Engineering, Nepti, Ahilyanagar, Maharashtra, India*

[2,3,4] *BE Scholar, Department of Artificial Intelligence and Data Science Engineering, Shri Chhatrapati Shivaji Maharaj College of Engineering, Nepti, Ahilyanagar, Maharashtra, India*

Corresponding Author: virajdarandale1@gmail.com

Received: 10.09.2026	Revised: 18.09.2026	Published: 31.09.2026

ABSTRACT

Investors and researchers must combine prices, company results, filings, news and economic data that arrive in different forms and at different speeds. Simple screeners apply a few fixed filters, and many machine-learning tools predict prices without explaining their answers. This paper describes the architecture of AI-SMIS, an explainable framework for researching Indian listed companies, and fixes in advance the protocol by which it will be tested. The framework combines a point-in-time data pipeline that records the source and public-release date of every value; nine analysis modules (fundamental, technical, valuation, sector, market regime, sentiment, events, risk and anomaly); four machine-learning tasks explained with SHAP; an additive intelligence score whose changes can be traced exactly to its components; and a tool-grounded research assistant that answers only from retrieved evidence. For reproducibility, the paper specifies the study period (April 2011 to March 2026), nine expanding-window walk-forward folds with purging and embargo, label definitions, default score weights, rule-based market-regime labelling, hyper-parameter search ranges, and the models used for sentiment, embeddings and the language model. It also pre-specifies a six-step feature-ablation ladder, four hypotheses with quantitative decision rules and named statistical tests, and an annotated evaluation of the research assistant against a no-retrieval baseline. No empirical results are reported here; all example values are illustrative and are labelled as such. The framework supports research and decision making; it is not a trading system and does not give investment advice.

Keywords: *stock market intelligence; explainable AI; multi-factor analysis; retrieval-augmented generation; point-in-time data; walk-forward validation; evaluation protocol; Indian equity market*

I. INTRODUCTION

Anyone who has tried to research a stock knows how scattered the information is. There are price charts, quarterly results, annual reports, shareholding details, news stories, company announcements and a constant flow of opinion. A company's condition depends on many things at once: how healthy its finances are, how fast it is growing, whether its shares look cheap or expensive, how its sector is doing, what the wider market is like, and what has just happened to it. No single number, whether it is the price-to-earnings ratio, a momentum indicator or last month's return, can capture all of this.

In practice, though, these questions are usually studied one at a time. A fundamental screen may pick out strong businesses while the market is falling apart. A chart-based tool may spot momentum in a company with weak finances. A news tool may tell you a headline is positive without knowing how that company's numbers look. The analyst is left to join the pieces together by hand, which takes a lot of time and is easy to get wrong.

Machine learning has been used heavily on this problem. Large studies show that flexible models can find useful patterns in many company characteristics at once [1], and deep-learning surveys describe a wide range of stock-prediction work [2], including studies on the Indian market [3], [4]. Most of these systems, however, are built to do one job, which is to predict a price or a return. They hand back a score or a buy/sell label and say little about

 © 2026 VT International Press. All rights reserved. E-ISSN:3139-7026

which factors drove it, how good the data was, or how far the answer can be trusted. For research and decision support that is not enough. A person wants to know what is happening, why it is happening, which stocks and sectors are affected, what history says about similar situations, and how reliable the analysis is [5], [6].

Financial data also brings three special difficulties. The first is fragmentation: prices, filings, news and documents come from different places in different formats. The second is that financial models are easy to test badly. If a model is allowed to see information that was not public at the time, if the stock universe is reconstructed with hindsight, or if training and test data are mixed at random, the results can look excellent and then fail in real use [7], [8], [9]. The third is that language models can write fluent summaries of financial text but sometimes state things that are not supported by the source [10]; recent finance-specific evaluations show that this happens often, even with retrieval [11], [12]. In a field where people act on what they read, that is a serious problem.

This paper describes **AI-SMIS**, an AI-based Stock Market Intelligence and decision-support framework for Indian listed companies. It is built around one central question: *can combining company financials, price behaviour, valuation, sentiment, market conditions, events and machine-learning output give a fuller and easier-to-understand picture of a stock than any one of these used alone?* The paper is deliberately a system-design and evaluation-protocol paper: it specifies the architecture and fixes, before any test data are examined, exactly how the central question will be answered. It reports no performance results; those will be reported in a companion empirical paper that follows this protocol. The paper makes five contributions:

- a layered architecture that brings market, financial, text and reference data together into one explainable profile for each stock;
- a data design that records the source and public-release date of every value and uses point-in-time index membership, so that models only use what was genuinely known at the time;
- an additive score that can be taken apart into its components, with a published default weight vector, confidence levels, data-quality notes, and an exact explanation of why a score changed;
- an AI research assistant that works through controlled analysis tools and answers only from evidence it has retrieved, with its models, retrieval settings and evaluation procedure fully specified; and
- a complete, reproducible evaluation protocol — study period, folds, labels, hyper-parameter ranges, hypotheses, named statistical tests and quantitative decision rules — to be registered before evaluation.

The rest of the paper is organised as follows. Section II reviews related work, including studies on the Indian market and recent work on language-model agents in finance, and identifies the gap. Section III explains the research design, data, features, models, validation and the reproducibility specification. Section IV describes the system. Section V presents illustrative outputs and the evaluation protocol. Section VI discusses implications and limitations, Section VII states data and code availability, and Section VIII concludes.

II. LITERATURE REVIEW

The framework draws on six areas of earlier work: research on what drives stock returns, machine learning for prediction, analysis of financial text, explainable AI, language-model assistants and agents for finance, and studies of the Indian market. We look at each in turn and then show where AI-SMIS fits.

A. What Drives Returns, and Machine Learning on Top of It

Early asset-pricing research showed that differences in stock returns are linked to a small set of company traits. Fama and French [13] found that market exposure, company size and the book-to-market ratio help explain returns, and Jegadeesh and Titman [14] showed that stocks that had recently done well tended to keep doing well for a while. These findings are the root of the multi-factor view used in this paper. Over time, though, researchers proposed hundreds of such factors, and Harvey *et al.* [15] warned that many of them may be false discoveries produced by testing too many ideas on the same data.

Machine learning was then brought in to handle the large number of signals. Gu *et al.* [1] compared linear models, regularised models, tree-based models and neural networks on a wide set of stock characteristics, and found that the non-linear methods did better, mostly because they could capture interactions between predictors. Fischer and Krauss [16] reported that LSTM networks beat simpler classifiers on S&P 500 stocks over a long test period. Jiang [2] surveyed the wider use of deep learning for stock prediction. Tree-based methods such as random forests [17] and gradient boosting [18], [19] remain strong choices for the kind of table-shaped data used here. For that reason, AI-SMIS tests logistic regression, random forest, XGBoost, LightGBM and a small neural network side by side instead of assuming that the most complex one wins.

B. Financial Text, Sentiment and Events

News and text carry real information about markets. Tetlock [20] showed that the negative tone of media coverage was linked to downward pressure on prices. Araci [21] adapted a language model specifically to financial text (FinBERT) and improved sentiment classification, and larger finance-focused models have since appeared, such as BloombergGPT [22] and the open-source FinGPT [23]. Many text tools, however, stop at a positive-or-negative label. AI-SMIS goes further: it records which company is involved, what kind of event happened, how the news

© 2026 VT International Press. All rights reserved.

reads, how big the likely impact is, over what time horizon, and how confident the system is. It then looks at how stocks behaved after similar events in the past.

C. Explainability, Confidence and Careful Testing

When people rely on a model, they need to understand it. Rudin [5] argues that explaining a black-box model after the fact can be misleading, and that clear, interpretable designs should be preferred for high-stakes decisions. Doshi-Velez and Kim [6] ask for interpretability to be tested in a rigorous way rather than claimed. SHAP [24] gives each input a fair share of credit for a prediction, and AI-SMIS uses it to show which factors pushed a score up or down. Reliability also depends on honest probabilities, so the framework checks calibration [25].

On testing, López de Prado [7] and Bailey *et al.* [8] explain how overlapping labels, information leaks and repeated backtesting make weak strategies look strong. They recommend time-ordered testing with purged and embargoed gaps between training and test periods, and this framework follows that advice. Survivorship bias — studying only the firms that still exist — inflates measured performance [9], so index membership must be taken as it was at each date. Market conditions also change over time. Hamilton [26] introduced the idea that an economy can switch between hidden states, while Pagan and Sossounov [27] and Lunde and Timmermann [28] gave rule-based ways of dating bull and bear markets. Section III-H explains which of these ideas is used for the regime features and why.

D. Retrieval-Augmented and Tool-Using Language Models in Finance

Retrieval-augmented generation (RAG) lets a language model look up relevant documents before it answers [29]. This reduces, but does not remove, the problem of made-up statements [10]. Agents that mix reasoning with calls to outside tools [30] go one step further. Since 2023 several language-model agents for finance have appeared. FinMem [31] gives a trading agent a layered memory of news, filings and prices, and TradingAgents [32] organises several language-model "analysts", "researchers" and "traders" into a simulated trading firm. Both are evaluated mainly by trading returns, and neither is designed to show the evidence behind each statement.

Evaluation work points to why this matters. On FinanceBench [11], an open-book question-answering benchmark built from company filings, strong models with a retrieval system still answered a large share of questions incorrectly or refused to answer. Kang and Liu [12] found serious hallucination in financial tasks and reported that retrieval and tool calls reduced it. FinBen [33] extended benchmarking to many financial tasks, including RAG and agent tasks, and RAGAS [34] offers reference-free measures of whether a RAG answer is faithful to its retrieved context. These results support the design used here: the language model should not produce numbers itself; it should call controlled analysis functions, collect what they return, explain it, and say "insufficient evidence" when there is none. They also show that such claims must be measured, which is why Section V-D specifies an annotated evaluation against a no-retrieval baseline.

E. Evidence from the Indian Market

Machine-learning studies on Indian equities have mostly focused on predicting the direction or level of an index or of a few stocks. Patel *et al.* [3] predicted the direction of the CNX Nifty, the S&P BSE Sensex and two large stocks from technical indicators and found that random forests performed well, especially when indicators were converted to trend signals. Selvin *et al.* [4] compared LSTM, RNN and CNN models for price prediction of NSE-listed companies. On text, Varghese and Mohan studied causal links between financial-news sentiment and Indian stock prices using Granger causality [35] and, with FinBERT scores, transfer entropy, finding that negative news has the stronger effect [36]. Balasubramanian *et al.* [37] used FinBERT on long news articles about NIFTY 50 companies to build equity portfolios, and Chaithra *et al.* [38] adapted an instruction-tuned language model with retrieval and market feedback to classify sentiment of NIFTY 50 news. Together these studies show that technical, fundamental and text signals carry information in the Indian market, but each looks at one or two kinds of information, a few stocks or an index, and none combines point-in-time data, factor-level explanation, confidence reporting and evidence-grounded generation in one framework that is tested with a pre-specified ablation.

F. The Gap

Table 1 summarises how typical tools in each group compare, with cited examples for each group. Each solves part of the problem, but none brings together multi-factor integration, clear data lineage, factor-level explanations, confidence reporting, evidence-based text generation, historical comparisons and portfolio or thesis tracking in one framework that can be measured. AI-SMIS is designed to fill that gap. Just as important, it includes a way to test whether every extra type of information actually improves results.

Table 1. Comparison of typical stock-analysis tools and the proposed framework, with representative examples.

Capability	Rule-based factor screens	Price-prediction ML	News / sentiment studies	LLM assistants and agents	AI-SMIS (designed)
Representative examples	Factor sorts [13], [14]	[1], [16], [3], [4]	[20], [21], [36], [37]	[22], [23], [31], [32]	This paper
Multi-domain integration	No (few filters)	Partial (mostly price or characteristics)	No (text only)	Partial (agents combine sources)	Yes

© 2026 VT International Press. All rights reserved.

Point-in-time data lineage	Partial	Varies; rarely documented	Varies	No	Yes
Factor-level explanation	Partial (rules)	Rarely	Rarely	Natural-language only	Yes (SHAP and additive score)
Confidence / data-quality report	No	Rarely	Rarely	No	Yes
Evidence-grounded generation	No	No	No	Partial; hallucination reported [11], [12]	Yes (tool-grounded, with abstention)
Historical analogues & event impact	No	No	Partial	No	Yes
Portfolio & thesis monitoring	Partial	No	No	Trading decisions only	Yes (no order execution)

Note. Ratings summarise the cited representative works and typical tools in each group; they are not a complete survey. The AI-SMIS column describes design intent, which the protocol in Section V is meant to test.

III. METHODOLOGY

The study combines two approaches. First, it uses design science: we build an artefact, which here is the intelligence framework. Second, it uses controlled experiments: once the artefact exists, its parts are tested under identical conditions so the effect of each kind of information can be isolated. This paper covers the design and fixes the experimental protocol; the experiments themselves are reported separately.

A. Research Design

The work is organised around five research questions. **RQ1:** does combining company financials, price behaviour, valuation and sentiment improve stock ranking compared with using one of them alone? **RQ2:** does information about market conditions make models more stable across different market environments? **RQ3:** do news and events add useful context beyond structured numbers? **RQ4:** can explainable AI make machine-learning assessments clear enough for research and decision support? **RQ5:** can an evidence-based AI assistant find, combine and explain information without unsupported claims?

These questions lead to four hypotheses (H1–H4), which are stated with their decision rules in Section V. The framework is deliberately not built to produce a simple BUY or SELL signal. Instead it tries to answer five plain questions about any stock: what is happening, why, which stocks and sectors are affected, what history suggests, and how reliable the analysis is.

B. Data Sources, Stock Universe and Survivorship

Four kinds of data are used. *Market data* means daily price candles with volume, index and sector-index series (including total-return indices), India VIX and corporate actions, collected through a broker market-data API [39]. *Financial data* means income, balance-sheet and cash-flow items, earnings per share, return ratios (ROE and ROCE), margins, debt, shareholding, and quarterly and annual results. Official exchange and regulator filings [40], [41] are treated as the source of truth instead of third-party websites. *News and alternative data* means exchange corporate announcements and licensed English-language financial news (Section III-H). *Reference and macro data* means a master list of securities (symbol, ISIN, exchange, sector, industry, index membership history and listing status), a calendar of company events, and economic series such as interest rates, inflation, the USD/INR rate, commodity prices and global indices.

The market-data connection sits behind a provider-neutral interface, so a data provider can be changed later without rewriting the system. The stock universe grows in stages. **Stage 1**, the subject of the protocol in this paper, is the NIFTY 50. Stage 2 is the NIFTY 500, and Stage 3 is the wider NSE and BSE market if data and infrastructure allow.

Survivorship bias. The universe on each scoring date is the set of index constituents *on that date*, rebuilt from the historical constituent changes published by the index provider, not today's constituents back-filled into the past [9]. Stocks that leave the index remain in the data for every date on which they were members, and companies that are later delisted, merged or suspended are kept with their returns up to the last traded price; the outcome window for such a stock ends on its last trading day and the remaining days earn the benchmark return. Symbol changes and mergers are tracked through the ISIN so that a renamed company is not treated as a new one. The share of constituent-months affected by delisting or index exit will be reported.

C. Data Engineering, Provenance and Point-in-Time Correctness

Data from all sources passes through adapters into a *raw zone*, where original files are saved exactly as received. It is then parsed, cleaned and checked, and stored in a *processed zone* in a relational database (Fig. 1). Keeping the raw copy matters. If a parsing mistake is found months later, the original filing can simply be processed again instead of being collected again. Documents such as results filings go through a parser that turns a sentence like "revenue from operations was …" into a structured record with the company, metric, period, value, unit, release date and source.

© 2026 VT International Press. All rights reserved.

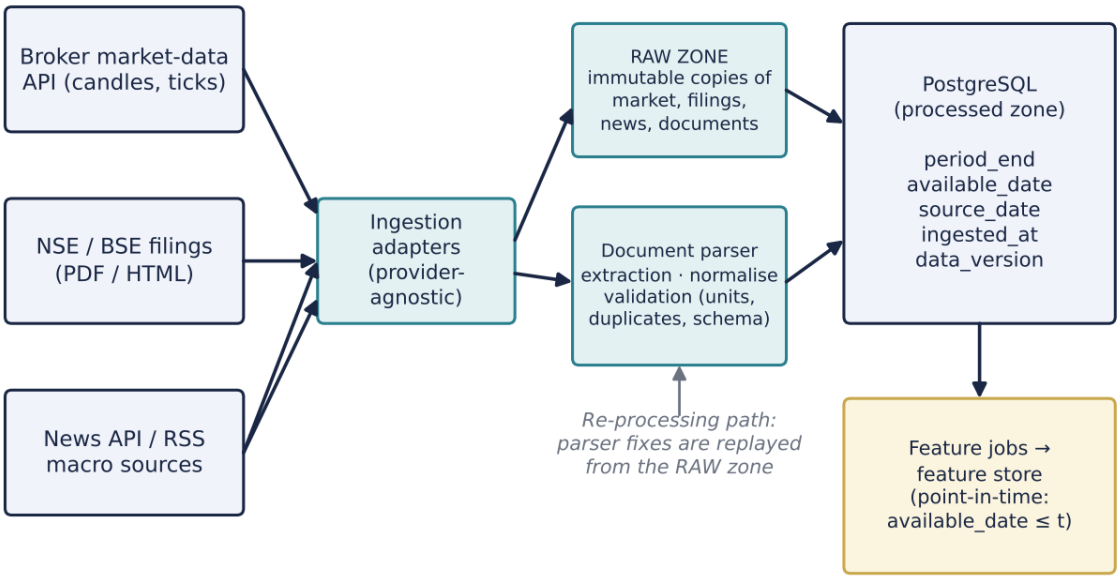


Figure 1. Data-engineering pipeline with raw and processed zones and provenance fields.

Every record carries a small set of dates and labels: *period_end* (the period the value describes), *available_date* (when it became public), *source_date*, *ingested_at* and *data_version*. Predictions also carry a *model_version*. The rule that follows is simple. A model predicting for a given day may use only information that was public on or before that day. When the exact release time of a filing is unknown, the value is treated as available on the next trading day. This rule, together with point-in-time index membership (Section III-B), is the main protection against look-ahead bias, information leakage and survivorship bias.

Features are calculated by scheduled jobs and saved in feature tables. Price-based features are updated after the market closes, and company-financial features are updated when a new filing arrives. Dashboards and models then read ready-made values instead of recalculating everything each time. Each dataset also shows how fresh it is (for example "live", "two days old" or "latest filing"), and this freshness feeds into the confidence estimate described later.

D. Feature Engineering

Nine families of features are built, one for each analysis module of the engine except anomaly detection, which produces flags rather than model inputs, plus a historical-analogue family (Table 2). Company-financial features are mostly expressed as trends, such as revenue growth, margin change and ROCE direction, because a trend says more than a single level. Valuation features are expressed relative to peer companies and to the company's own past. For banks and other financial companies, for which ROCE and debt-to-equity are not meaningful, return on assets and return on equity are used instead, and these firms are compared only within their own sector.

Table 2. The nine feature families used by the stock intelligence engine.

Family	Representative features	Refresh
1. Fundamental	ROE, ROCE (ROA for financials), margins, debt ratios, free-cash-flow trend, promoter and institutional holding, pledged shares	On filing
2. Growth	Revenue, operating-profit and EPS growth (quarter-on-quarter, year-on-year, trailing twelve months) and their acceleration	On filing
3. Technical	EMA 9/15/20/50/200, RSI, MACD, ADX, ATR, Bollinger Bands, volume ratio, support/resistance, 52-week position, 12-1 month momentum, relative strength	Daily
4. Valuation	P/E, P/B, EV/EBITDA, PEG, gap to sector peers and to own five-year median, gap to discounted-cash-flow value and its sensitivity	Daily / on filing
5. Sentiment	Article sentiment (FinBERT), 30-day decayed sentiment, news volume, sentiment dispersion	Intraday
6. Event	Event type (12-type taxonomy), days since event, expected impact, time horizon, confidence, event score	On announcement
7. Sector	Sector relative strength, breadth, sector valuation and earnings revision	Daily
8. Market regime	Trend state (bull, bear, sideways) and volatility state (high, normal, low) from rules in Table 6	Daily
9. Risk & historical	Realised volatility, beta, drawdown, earnings variability; similarity to past situations and what followed after 5, 20 and 60 days	Daily

© 2026 VT International Press. All rights reserved.

E. Models, Scoring and Analysis Methods

Machine-learning tasks. Instead of one price predictor, four separate tasks are defined. Task A classifies company quality, Task B classifies the financial trend as improving, stable or deteriorating, Task C estimates the probability that a stock will beat a benchmark over a fixed horizon, and Task D sorts stocks into low, medium and high risk. The exact label definitions, horizons and class balance are given in Table 4. The candidate models are logistic regression, random forest [17], XGBoost [18], LightGBM [19] and a small multilayer perceptron, built with standard libraries [42], with search ranges in Table 7. All are trained and tested on the same splits and compared in the same way.

The intelligence score. Each stock receives eight component scores, from 0 to 100: fundamentals, growth, valuation, technical condition, sector, sentiment, risk and machine-learning outlook. Each component is the cross-sectional percentile rank of its family composite on the scoring date, so that 100 is the best stock in the universe; the risk score is inverted so that a higher score means lower risk. The components are combined into one overall score as a weighted sum with non-negative weights that add up to one. The default (Balanced) weights and the four user presets (growth, value, quality, low risk) are listed in Table 5. Because the overall score is a weighted sum, a change in the overall score can be traced exactly to its parts: the contribution of component i is its weight times its change. That is how the "why did the score change?" feature works (Fig. 7b). It reads the real changes in the underlying data and does not ask a language model to guess.

Valuation. Relative valuation uses P/E, P/B, EV/EBITDA, PEG and comparison with peers. Intrinsic valuation uses discounted cash flow, which estimates what a company's future cash flows are worth today [43]. Because the result is sensitive to the discount rate and the long-term growth rate, the system shows a range of scenarios and how sensitive the answer is, instead of one "fair value".

Market regime, sectors and events. The market is labelled with a trend state (bull, bear or sideways) and a volatility state (high, normal or low) by the transparent rules in Table 6, which use only data available at each date. Sectors are ranked on relative performance, financial strength, momentum, breadth and valuation, so users can drill down from sector to industry to company. When an event is detected, the event-impact engine finds similar past events and reports how many were found, the typical 5-day and 20-day price change, and how widely the outcomes varied. A historical-analogue engine describes a stock's current state (growth, profitability, valuation, momentum, volatility, sector strength, market regime and sentiment) and searches for similar past situations, showing what followed after 5, 20 and 60 days. The result is historical context, not a forecast.

Unusual behaviour. The system flags unusual movements in price, volume, financial ratios, margins, earnings or ownership. It uses robust z-scores (distance from the 60-day median in units of median absolute deviation, flagged above 3.5), together with Isolation Forest [44] and autoencoders. A flagged item is shown with its score and the supporting evidence. It is not automatically called good or bad, because an anomaly only means something deserves a closer look.

F. Validation Design

Models are tested with expanding-window walk-forward validation (Fig. 2). Each fold has a training period, a validation period used for choosing settings, and a later out-of-sample test period that the model has never seen. Training samples whose label window overlaps the next block are purged, and a further embargo equal to the label horizon separates the blocks, so that labels that look into the future cannot overlap [7]. After settings are chosen on the validation block, the model is refitted on training plus validation data and then scored once on the test block. The study period, number of folds and window lengths are given in Table 3. Results are reported by fold and by market regime, because the goal is stability and not just a good average.

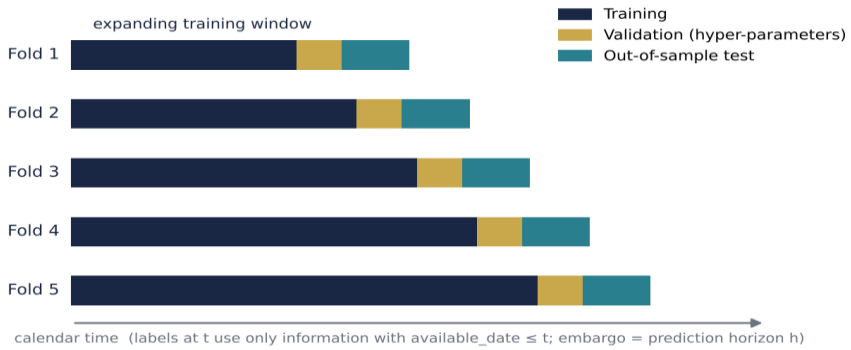


Figure 2. Expanding-window walk-forward validation with validation and out-of-sample test blocks (schematic; the protocol uses nine folds, Table 3).

To see what each type of information adds, six experiments are run on identical splits, labels and metrics (Fig. 3; Table 9). Experiment A uses company financials (fundamental and growth families) only. B adds technical features. C adds valuation. D adds sentiment. E adds sector, market-regime and event features. F is the full system,

© 2026 VT International Press. All rights reserved.

adding the risk-and-historical family, machine-learning scoring and calibration. The same ladder is run for every candidate model, which also shows why one model beats another.

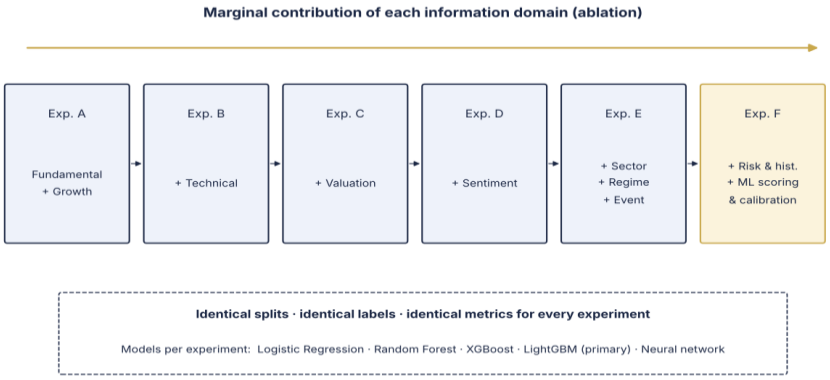


Figure 3. Step-by-step comparison used to isolate what each kind of information contributes.

G. Explainability, Confidence and Traceability

For tree-based and similar models, each prediction is explained with SHAP [24]. SHAP splits a prediction into one contribution per feature, so the platform can show the main positive factors (for example faster revenue growth, improving ROCE and strong relative strength) and the main negative factors (for example high valuation and rising volatility) instead of a bare number.

Each model output also comes with a confidence profile. It takes into account the model's probability, whether different models agree, the quality and freshness of the data, how much history is available, how similar the current market is to the past, and how well calibrated the model is [25]. If models disagree or the current market type is rare in the training data, the output is marked as low confidence and the reason is given. This avoids giving model estimates more precision than they deserve. Finally, every result can be traced step by step, from the stock score to its component scores, to the features, to the source data, and to the original document and its timestamp.

H. Reproducibility Specification

This subsection fixes every setting needed to reproduce the Stage-1 experiments. Tables 3 to 8 give the study design, the label definitions, the score weights, the market-regime rules, the hyper-parameter search ranges and the text and retrieval configuration. Any later change will be reported in a "deviations from protocol" table in the empirical paper.

Table 3. Study design for Stage 1 (NIFTY 50).

Item	Specification
Universe	Point-in-time NIFTY 50 constituents on each scoring date, including later exits and delistings (Section III-B)
Data period	1 April 2010 to 30 June 2026; the first year is used only to warm up features (e.g., 200-day averages, trailing growth)
Sample period (labels)	Scoring dates from April 2011 to March 2026; outcome windows run to June 2026
Scoring frequency	Last trading day of each month, using information available at that day's close
Walk-forward folds	9 expanding-window folds. Fold k (k = 1,…,9): training from April 2011 to March of year 2015+k; validation = the next financial year; test = the financial year after that. Test years run from FY2017-18 to FY2025-26 (108 test months)
Purge and embargo	Training samples whose outcome window overlaps a later block are removed, plus an embargo equal to the task horizon (63 trading days for Tasks C and D; 12 months plus a 60-day filing lag for Tasks A and B)
Primary horizon (Task C)	63 trading days (about three months); secondary horizons 21 and 126 trading days
Benchmark	NIFTY 50 Total Return Index (Stage 2: NIFTY 500 TRI); non-ML baselines: equal-weight universe, 12-1 month momentum rank, and low P/B rank
Primary model for hypothesis tests	LightGBM, fixed in advance to avoid picking the best model after seeing the results; other models are reported descriptively
Portfolio metrics	Equal-weight top-10 portfolios formed monthly and held 3 months as overlapping sub-portfolios [14]; 25 bp one-way cost (sensitivity 10–50 bp); used only to describe financial usefulness, not as a trading claim
Random seeds	Seed 42 for search; final models averaged over five seeds (42–46)

Table 4. Label definitions for the four machine-learning tasks.

Task	Label definition (features use only data available on the scoring date)	Class balance and primary metric
A. Quality	Tercile (within the month's cross-section) of a quality composite measured over the next four reported quarters: mean ROCE (ROA for financials), operating	≈ 1/3 per class by construction; macro-F1

© 2026 VT International Press. All rights reserved.

	cash flow / net profit, negative debt-to-equity and negative earnings variability, each z-scored within sector	
B. Financial trend	Change in trailing-twelve-month operating-profit growth over the next four quarters: improving if it rises by at least 5 percentage points, deteriorating if it falls by at least 5 points, otherwise stable	Imbalanced; class shares reported per fold; class-weighted loss; macro-F1
C. Outperformance	1 if the stock's 63-day total return minus the NIFTY 50 TRI return over the same window is above zero, else 0; secondary label: top quintile of excess return	Near-balanced; share of positives reported per fold; NDCG@10, ROC-AUC
D. Risk	Tercile (within the cross-section) of realised annualised volatility of daily log returns over the next 63 trading days: low, medium, high	≈ 1/3 per class by construction; macro-F1

Table 5. Default (Balanced) and preset weight vectors for the overall score.

Component	Balanced (default)	Growth	Value	Quality	Low risk
Fundamentals	0.20	0.15	0.20	0.35	0.20
Growth	0.15	0.30	0.05	0.10	0.05
Valuation	0.15	0.05	0.35	0.10	0.10
Technical	0.10	0.15	0.05	0.05	0.05
Sector	0.10	0.10	0.10	0.05	0.10
Sentiment	0.10	0.05	0.05	0.05	0.05
Risk (higher = lower risk)	0.10	0.05	0.10	0.20	0.35
ML outlook	0.10	0.15	0.10	0.10	0.10
Sum	1.00	1.00	1.00	1.00	1.00

Note. Weights were set by the authors before any test data were examined, to reflect the emphasis of each investing style; they were not fitted to returns, which avoids adding a further layer of data snooping [15]. Their sensitivity will be reported by recomputing all results with equal weights (0.125 each) and with each weight moved by ±0.05 (renormalised). Data-driven weights, if explored, will be learned within training folds only and reported as exploratory.

Table 6. Market-regime labelling rules (computed daily on the NIFTY 50 TRI; NIFTY 500 TRI in Stage 2).

State	Rule (inputs available at the close of day t)
Bull	Index close above its 200-day simple moving average AND 126-day index return above +5%
Bear	Index close below its 200-day simple moving average AND 126-day index return below −5%
Sideways	Neither of the above
Confirmation	A new trend state is adopted only after its condition holds for 5 consecutive trading days (prevents flip-flopping)
High / normal / low volatility	21-day average of India VIX above the 70th percentile, below the 30th percentile, or between them; percentiles come from an expanding window that ends at day t−1
Combined regime	Trend × volatility (up to 9 cells); for H2, cells with fewer than 12 test months are merged into their trend state
Robustness check	Two-state Markov-switching model [26] on weekly index returns, re-estimated in each fold on training data only, using filtered (not smoothed) probabilities; agreement with the rule-based labels is reported

Note. Labels are stored with their available_date and are never revised afterwards. Turning-point methods [27], [28] need future prices to confirm a peak or trough, so they are used only to describe the sample after the fact, never as model inputs.

Table 7. Hyper-parameter search ranges (random search, 60 trials per model per fold, on the validation block only).

Model	Search space
Logistic regression	L2 penalty; C ∈ [10^{-3}, 10^{2}] (log-uniform); class_weight ∈ {none, balanced}; standardised inputs
Random forest	n_estimators ∈ [200, 1000]; max_depth ∈ [3, 12]; min_samples_leaf ∈ [5, 50]; max_features ∈ {sqrt, 0.3, 0.5}
XGBoost	learning_rate ∈ [0.01, 0.1] (log); max_depth ∈ [2, 6]; n_estimators ≤ 1000 with early stopping after 50 rounds; subsample and colsample_bytree ∈ [0.6, 1.0]; reg_lambda ∈ [0, 10]
LightGBM	learning_rate ∈ [0.01, 0.1] (log); num_leaves ∈ [8, 64]; min_child_samples ∈ [20, 200]; feature_fraction and bagging_fraction ∈ [0.6, 1.0]; lambda_l2 ∈ [0, 10]; early stopping after 50 rounds
Multilayer perceptron	1–3 hidden layers of 32–128 units; dropout ∈ [0, 0.3]; weight decay ∈ [10^{-5}, 10^{-3}]; Adam, learning rate ∈ [10^{-4}, 10^{-2}]; early stopping (patience 20 epochs)
Selection objective	NDCG@10 for Task C; macro-F1 for Tasks A, B and D. Probabilities are calibrated by Platt scaling on the validation block

Table 8. Text, event and retrieval configuration.

Component	Specification
Text sources and language	NSE and BSE corporate announcements and filings (primary); licensed English-language Indian business news. English only in Stages 1–2; Hindi and Marathi news are out of scope (Section VI)
Sentiment model	FinBERT [21] (public ProsusAI/finbert checkpoint) on headline plus first two sentences; score = P(positive) − P(negative). Before use, it is checked on 1,000 Indian headlines labelled by two annotators; if macro-F1 is below 0.70 it is fine-tuned on a separate labelled set
Event extraction	Exchange announcement categories and rule patterns mapped to a 12-type taxonomy (results, dividend, buyback, bonus/split, merger/acquisition, rating change, management change, promoter pledge, order win, regulatory/legal action, fund-raising, other); the language model is used only for unmatched items, and its output must be one of the 12 types
Language model (agent)	Llama 3.1 8B Instruct [45], open weights, served locally; temperature 0; maximum 1,024 output tokens; prompt templates versioned in the code repository

© 2026 VT International Press. All rights reserved.

Embedding model	bge-base-en-v1.5 (768 dimensions) [46]
Vector store	PostgreSQL with the pgvector extension; HNSW index; cosine similarity
Chunking	Split at section headings, at most 400 tokens with 50-token overlap; each table kept as its own chunk with its header row; metadata: company, ISIN, document type, period_end, available_date, page
Retrieval	Hybrid BM25 + dense search combined by reciprocal-rank fusion; top 8 chunks; only chunks with available_date on or before the question's as-of date
Evidence guard	Every sentence must cite a chunk or tool-output identifier; numbers are checked against tool outputs after unit normalisation; sentences that fail are removed; if no supporting item remains, the answer is "insufficient evidence"

IV. SYSTEM DESIGN AND IMPLEMENTATION

A. Layered Architecture

AI-SMIS has six layers (Fig. 4): data sources, data engineering, the stock intelligence engine, the advanced AI layer, application services, and presentation. Each layer only talks to the one next to it. The first version is built as a modular monolith, which means one application with clearly separated modules. This avoids the extra complexity of micro-services while still keeping the parts easy to change.

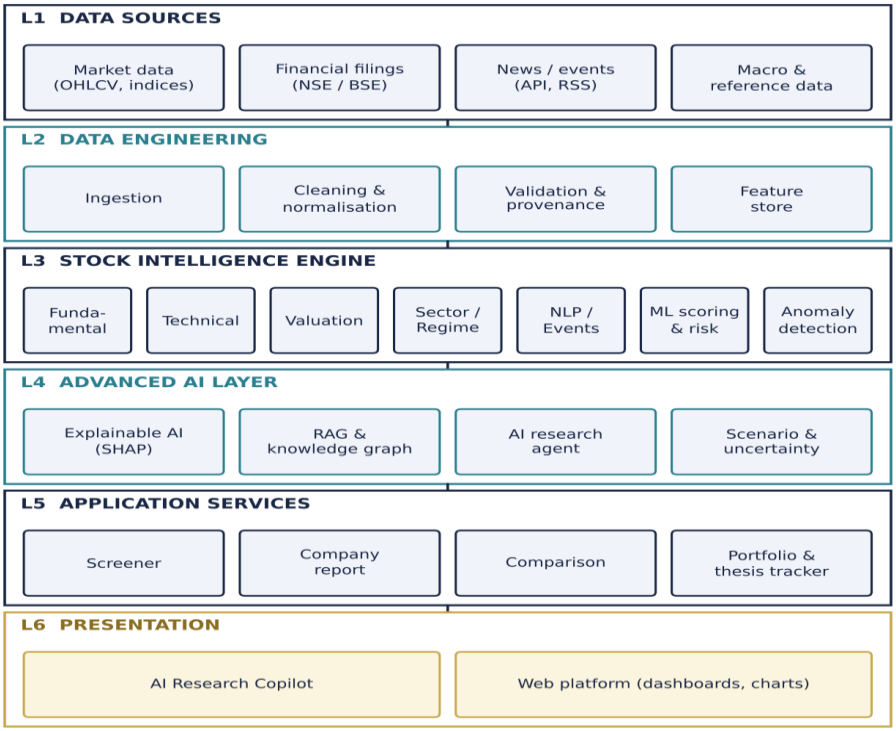


Figure 4. Six-layer architecture of the AI-SMIS framework.

B. Stock Intelligence Engine and Application Services

The engine holds the nine analysis modules described in Section III — fundamental, technical, valuation, sector, market regime, sentiment, events, risk and anomaly — together with the machine-learning and historical-analogue components. On top of it sit the tools a user actually touches:

- **Intelligent screener.** Users combine many criteria, for example market capitalisation, ROCE, revenue growth, debt-to-equity, an RSI range, price against the 200-day average and the ML score. The screener ranks the matches instead of only filtering them.
- **Company report.** A research report is generated for each company from the stored data.
- **Comparison.** Several companies are compared side by side, and the system explains why the ranking changes with the criteria chosen.
- **Portfolio analyser.** It studies a portfolio the user keeps, looking at sector exposure, concentration, correlation, volatility, drawdown and valuation risk. It analyses holdings and does not place trades.
- **Thesis tracker.** The user states why they like a stock (for example growth above a threshold, ROCE above a threshold, positive free cash flow). The system checks these conditions as new data arrives and reports a simple health status such as "4 of 5 conditions met".
- **Timeline.** Each company has a history of financial changes, news, valuation, technical state and score, which shows how its investment profile has changed over time.

© 2026 VT International Press. All rights reserved.

C. Generative-AI Layer: RAG, Knowledge Graph and Research Agent

Annual reports, quarterly results, investor presentations and call transcripts are parsed, split into chunks, converted into embeddings and stored in a vector database for retrieval, using the models and settings in Table 8. Numbers extracted from the same documents also feed the financial analysis. A knowledge graph records how companies, sectors, industries, competitors, promoters, indices, events and themes relate to each other. This makes it possible to answer relationship questions that a plain table of stocks cannot, for example which competitors of a company are also affected by the same event.

The research agent sits in the middle and directs the work (Fig. 5). It first works out what the user is asking. It then picks from a fixed set of typed tools, such as stock search, financial history, valuation, technical analysis, market regime, events, historical analogues, score-change explanation and scenario runs. It collects what those tools return, and only then asks the language model to write the answer. Generated reports keep facts, model outputs, interpretations and AI-written explanations clearly apart. Each statement is linked to a metric, a period, a source and a date. If the evidence is not enough, the system says exactly that.

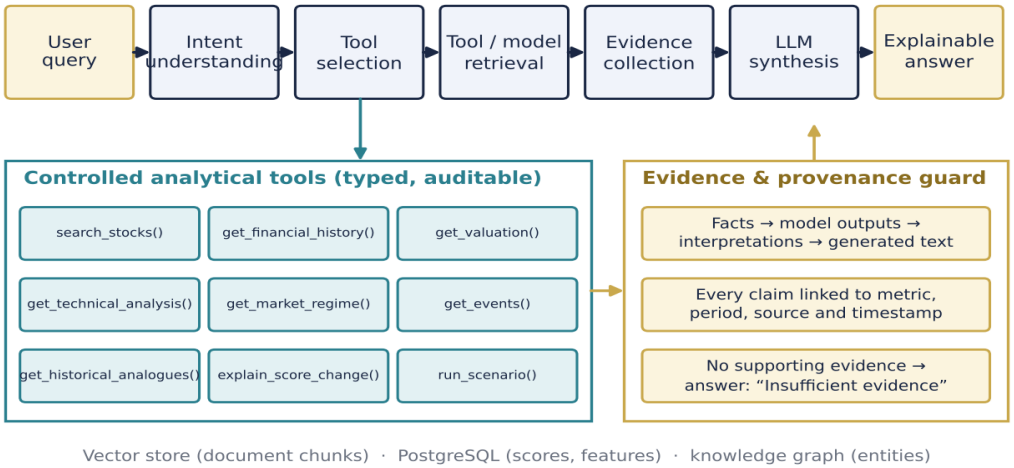


Figure 5. Tool-grounded research-agent workflow with an evidence and provenance guard.

A scenario engine lets the user try "what if" assumptions, such as lower growth, squeezed margins, higher interest rates or a higher discount rate. It recalculates revenue, profit, free cash flow, valuation and risk, and shows the outcome as one possible scenario and not as a prediction.

D. A Typical User Journey

Figure 6 shows the path a user is expected to follow. They start with the market overview and the current market regime, move to the strongest sectors, and use the screener to shortlist companies. After choosing a stock they see its full profile, ask the system to explain the score, look at similar historical situations, and finally ask the research assistant follow-up questions in plain language, such as "what changed in the last quarter?" or "show me companies with strong fundamentals but weakening charts". Each step answers the next natural question, so the user does not have to jump between separate tools.

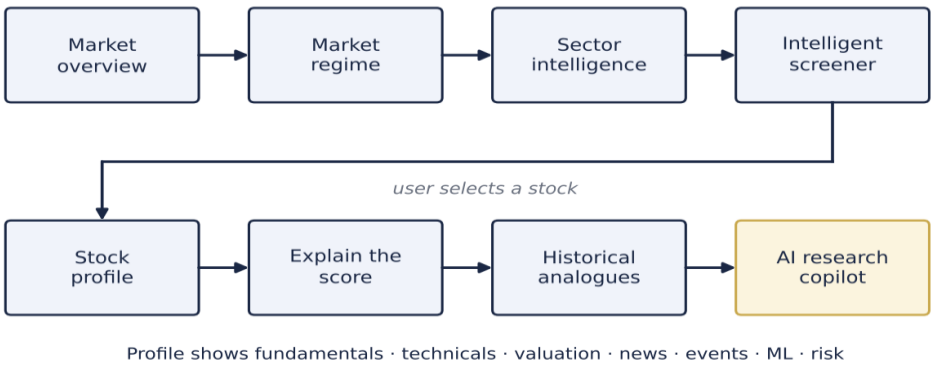


Figure 6. Expected user journey through the platform.

© 2026 VT International Press. All rights reserved.

E. Technology Stack and Data Model

The backend is written in Python with FastAPI. Data handling uses pandas and NumPy. Modelling uses scikit-learn, XGBoost, LightGBM and SHAP, and PyTorch is used for the neural-network experiments. PostgreSQL stores companies, instruments, prices, financial periods and metrics, index-membership history, news and events, documents and chunks, feature tables, predictions, model versions, experiments, scores and their history, anomalies, analogues, portfolios and theses; its pgvector extension serves as the vector store. Redis handles caching, and Neo4j is an option for the knowledge graph. The language model runs locally through an open-weights inference server. The frontend uses HTML, Tailwind CSS and TypeScript with interactive financial charts, and Docker is used for deployment. Because every score, prediction and answer is linked to a data version and a model version, results can be reproduced and checked later.

V. ILLUSTRATIVE OUTPUTS AND EVALUATION PROTOCOL

This paper presents the framework and its test plan; it does **not** report performance results, and research questions RQ1–RQ5 and hypotheses H1–H4 therefore remain open here. They will be answered in a companion empirical paper that runs this protocol on the Stage-1 (NIFTY 50) data, so that no result is quoted before it has been measured and no analysis choice is made after the results are seen. This section shows what the system's outputs look like, lays out the experiments, and states how each hypothesis will be judged.

A. What the Output Looks Like (Illustrative)

Figure 7a shows the format of the score breakdown for one stock using **illustrative values**: fundamentals 89, growth 91, valuation 74, technical 81, sector 85, sentiment 78, risk 84 and ML outlook 86. With the Balanced weights in Table 5 the overall score is 0.20×89 + 0.15×91 + 0.15×74 + 0.10×(81 + 85 + 78 + 84 + 86) = 83.95, displayed as 84. Figure 7b shows the score-change explanation for an example fall from 83.95 to 66.95 (displayed as 84 to 67). The component changes are valuation −40, sentiment −40, technical −30, risk −20, sector −10, ML outlook −10, fundamentals 0 and growth 0. Multiplying each change by its weight gives contributions of −6.0, −4.0, −3.0, −2.0, −1.0, −1.0, 0 and 0, which add up to exactly −17.0, so valuation is named as the main cause. All eight components are shown, including those that did not change. These panels show the output format only; they are not findings and will be replaced by real, clearly labelled outputs in the empirical paper.

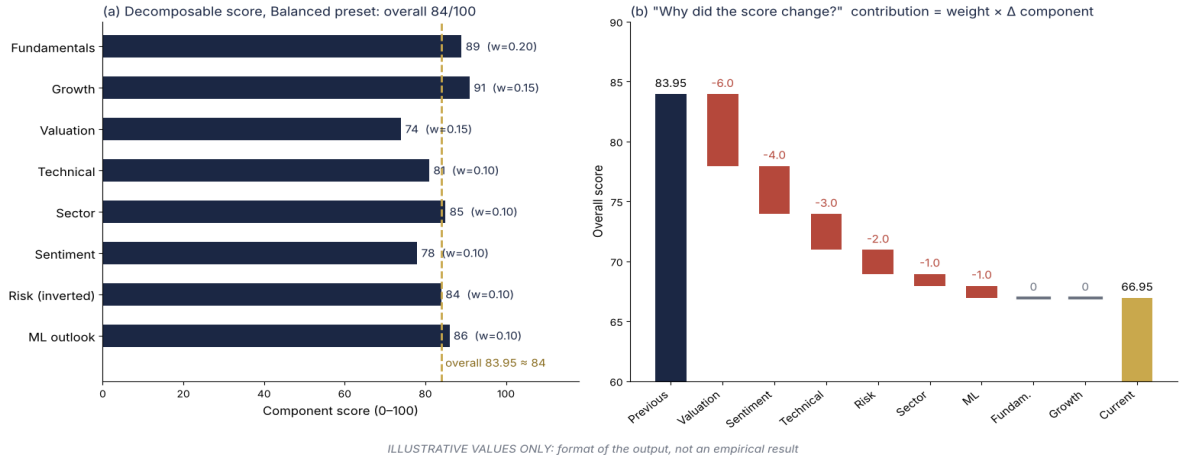


Figure 7. Illustrative output (not empirical results): (a) a score broken into its eight weighted components; (b) a score change traced back to its causes, where each contribution equals weight × component change.

B. Experiment Layout

Table 9. Planned experiments (each run for every candidate model on identical splits).

Exp.	Information used (feature families, Table 2)	Question answered
A	Fundamental + growth	Baseline: how well do financial characteristics alone rank stocks?
B	A + technical	Do price-chart signals add anything to the financials?
C	B + valuation	Does valuation context improve the ranking?
D	C + sentiment	Does text sentiment add information? (H4)
E	D + sector, market regime and events	Do market conditions and events improve stability? (H2, H4)
F	E + risk & historical, with ML scoring and calibration	Does the complete system beat every smaller set? (H1)

C. Hypotheses, Statistical Tests and Decision Rules

Measures. Ranking quality is measured on each monthly cross-section by NDCG@10 [47], with relevance equal to the quintile grade (0–4) of the realised excess return, together with top-10 precision and the Spearman rank correlation between predicted probability and realised excess return (rank IC). Classification quality is measured

© 2026 VT International Press. All rights reserved.

by accuracy, macro-F1, ROC-AUC, PR-AUC and expected calibration error. Financial usefulness is described by the forward excess return, hit rate and maximum drawdown of the top-10 portfolio (Table 3) and by stability across regimes. System quality is measured by data-processing and API response time, inference time, availability and error rate.

Inference. The unit of analysis is the month (108 test months). Because 63-day outcome windows overlap from month to month, confidence intervals are computed with the stationary bootstrap [48] (mean block length 3 months, 10,000 resamples) on paired monthly differences between experiments. A one-sided Wilcoxon signed-rank test [49] on the same paired differences is reported as a supporting check. Within each hypothesis, p-values and confidence levels are adjusted with the Holm procedure [50]; secondary metrics are controlled with the Benjamini–Hochberg procedure [51] at a 10% false-discovery rate and labelled exploratory. Table 10 states the decision rules.

Table 10. Research hypotheses, statistical tests and decision rules (primary model: LightGBM; primary task: C).

Hyp.	Statement	Comparison, measure and test	Decision rule (fixed in advance)
H1	A multi-factor model ranks stocks better than a single-family baseline.	Primary: Exp. F vs A; secondary: F vs B, C, D, E. Monthly ΔNDCG@10; stationary-bootstrap CI; Holm over the five comparisons	Accept if the Holm-adjusted 95% CI of mean ΔNDCG@10 (F − A) lies above zero AND F beats A in at least 6 of 9 folds
H2	Market-regime information makes models more stable across conditions.	Exp. E vs D. For each regime cell, mean monthly rank IC; dispersion = standard deviation across cells. Paired stationary-bootstrap test of the difference in dispersion	Accept if the 95% CI of [SD(E) − SD(D)] lies below zero AND mean rank IC of E is not lower than D by more than 0.01 (non-inferiority margin) AND top-10 maximum drawdown is not worse by more than 2 percentage points
H3	Explainable AI gives stable and financially sensible feature-level explanations.	(i) Stability: mean pairwise Kendall's τ of global mean |SHAP| rankings of the top 15 features across the 9 folds. (ii) Sign check against an expected-sign table frozen before results. (iii) Blinded analyst review (protocol below); agreement by Cohen's κ [52]	Accept if $\tau \geq 0.60$ AND at least 80% of the top-10 features have the expected SHAP direction AND at least 75% of reviewed explanations are rated "consistent" by majority, with mean pairwise $\kappa \geq 0.60$
H4	Financial-text information adds useful context beyond structured data.	Exp. D vs C and E vs C (ranking); RQ5 assistant with vs without text sources (evidence support). Stationary bootstrap; Holm over the comparisons	"Clear gain" = mean ΔNDCG@10 ≥ 0.010 AND mean Δrank IC ≥ 0.010 with Holm-adjusted 95% CI above zero, OR an increase of at least 10 percentage points in the share of questions answered with fully supported evidence

Analyst review protocol (H3). Three reviewers who were not involved in building the system take part: one finance faculty member and two practitioners with at least three years of equity-research or portfolio-management experience. Each reviewer independently rates 40 local explanations (stratified at random across folds, tasks and predicted classes) and the global top-10 feature list of each task as "consistent", "inconsistent" or "unclear" with sound financial reasoning, using a written rubric. Reviewers do not see the model identity or the realised outcome. Agreement is measured with pairwise Cohen's κ [52], and an explanation counts as consistent when at least two of the three reviewers rate it so.

D. Evaluation of the Research Assistant (RQ5)

RQ5 is tested on a fixed question set of 240 questions about 30 NIFTY 50 companies, each with an as-of date: 48 questions in each of four types (single-fact lookup, multi-period comparison, cross-company comparison, and "why did this change?") and 48 deliberately unanswerable questions for which the corpus holds no supporting evidence. Gold evidence for each answerable question is written by the annotators before any system is run. Three systems use the same language model and the same questions: (a) a **no-retrieval baseline** (the model answers from its own knowledge), (b) **plain RAG** (retrieval but no analysis tools), and (c) the **full AI-SMIS agent**.

Each answer is split into atomic statements. Two trained annotators (postgraduate students in finance, trained with a written rubric) label every statement as supported, partly supported or unsupported by the cited evidence; a faculty member settles disagreements. Agreement is measured with Cohen's κ [52]; if κ is below 0.70 on the first 40 questions, the rubric is revised and those questions are re-annotated before the rest are scored. The measures are the share of supported statements, the share of unsupported claims, numeric accuracy, precision and recall of "insufficient evidence" on unanswerable questions, and response time; RAGAS faithfulness [34] is reported as a secondary automatic measure. Differences between systems are tested by a paired bootstrap over questions with Holm correction. The claim of grounded generation is accepted if the full agent's unsupported-claim rate is below both baselines (95% CI excluding zero) and is at most 5%.

One principle runs through the whole evaluation: predictive performance and practical usefulness are not the same thing. A model that scores slightly lower but is far more stable and easier to understand may well be the better choice.

© 2026 VT International Press. All rights reserved.

E. Evaluation Across Layers

Table 11 shows how the evaluation covers every layer of the system and not only the models, because a strong model on weak data is still a weak system.

Table 11. What is evaluated at each layer.

Layer	What is checked	Example measures
Data	Completeness, correct dates, duplicates, freshness, survivorship	Missing-value rate, share of records with valid available_date, data age, share of exited or delisted constituent-months retained
Models	Prediction quality, ranking quality, calibration, stability	ROC-AUC, PR-AUC, NDCG@10, rank IC, calibration error, result by regime
Explanations	Are the factors stable and sensible?	Kendall's τ of SHAP rankings across folds, sign agreement, analyst ratings and κ
AI assistant	Are answers supported by evidence?	Share of supported statements, unsupported-claim rate, abstention precision and recall
Platform	Speed, reliability and usability	API latency, availability, error rate, user ratings

VI. DISCUSSION

The framework changes the question that is usually asked. Instead of "which way will this price move?", it asks "how is the information about this stock and this market changing, and how sure can we be?" Four design choices follow from the earlier research and set it apart from typical tools.

First, it combines several kinds of information, because predictive signals are spread across many company traits and interact in non-linear ways [1]. At the same time, the step-by-step experiments protect against the problem of too many factors, since every new kind of information has to prove its worth [15]. Second, it records the release date of every value, uses point-in-time index membership and tests models with purged walk-forward validation. These steps address the weaknesses that make many financial machine-learning results hard to repeat [7], [8], [9]. Third, scoring is additive with published weights and explanations come from SHAP, so explainability is part of the structure and not an add-on. The score-change feature uses real changes in the data and not generated text, which is in the spirit of the call for transparent tools in high-stakes settings [5]. Fourth, the language model is only allowed to explain evidence that controlled tools have returned, and it must say "insufficient evidence" when there is none. Recent evaluations suggest this is a practical way to reduce made-up statements [29], [12], but whether it works here is an empirical question that Section V-D will answer.

For students and practitioners, the platform shortens the path from raw data to a reasoned view. A person can move from the market regime to a sector, from there to screened companies, to one company's profile, to an explanation of its score, and to similar situations in the past, all in one place. For researchers, the fully specified protocol offers a repeatable way to ask whether sentiment, market regime or event information really adds value in the Indian market, instead of taking it for granted.

Limitations and Threats to Validity

- **No results yet.** This paper is a design and protocol paper. The contribution of the framework cannot be judged until the experiments are run, and the hypotheses may be rejected.
- **Data.** Good, licensed financial and news data is costly. Historical news can be incomplete, and company reports come in many formats, which makes automatic extraction error-prone.
- **Small universe and power.** Stage 1 has about 50 stocks per month and 108 test months, so only fairly large differences between experiments can be detected; minimum detectable effects will be reported with the results, and Stage 2 (NIFTY 500) will test whether findings generalise.
- **Market behaviour.** Prices are noisy and keep changing character. Models may be unstable from one regime to another, and rare regimes are thin in the training data. Historical analogues show what happened before and do not promise what will happen next.
- **Bias and language.** FinBERT was trained mainly on non-Indian financial text and may inherit its biases; it is therefore checked on Indian headlines first (Table 8). Only English text is used, so news in Hindi, Marathi and other Indian languages is missed. Survivorship bias is addressed by point-in-time membership but depends on the completeness of historical constituent records.
- **Many comparisons.** Many feature sets and models are compared, so results are judged with pre-specified primary comparisons, Holm correction and a primary model fixed in advance, and will be confirmed on data released after the protocol is registered.
- **Generative AI.** Retrieval lowers the chance of unsupported statements but does not remove it, so evidence checks and human review are still needed.

© 2026 VT International Press. All rights reserved.

Responsible Use

AI-SMIS is a research and decision-support tool. It does not promise returns, it does not trade at high frequency or on its own, and it does not give personalised investment advice. In India, providing investment advice to clients is a regulated activity governed by the SEBI (Investment Advisers) Regulations [53], so any public or commercial version of the platform would need to be reviewed against those rules. Every screen should say clearly that its output is information for research and not a recommendation.

VII. DATA, CODE AVAILABILITY AND LICENSING

- **Market data** are obtained through the Upstox developer API [39] under its terms of use for research and are not redistributed.
- **Exchange data** (corporate announcements, filings and historical index-constituent changes) are obtained from NSE and BSE public sources under their terms of use; the raw files are not redistributed, but the scripts that download and process them will be shared.
- **News data** come from a licensed feed whose terms do not allow redistribution of article text; only derived daily sentiment and event scores per stock will be shared, if the license permits.
- **Code.** All source code — data pipeline, feature construction, models, evaluation scripts and prompt templates — will be released under the MIT license in a public repository on acceptance, with an archived, citable version; the repository address and archive identifier will be added to the final version.
- **Synthetic sample.** A synthetic dataset that follows the same database schema will be released so that the full pipeline can be run end to end without licensed data.
- **Registration.** The analysis plan in Sections III-H and V, including the expected-sign table for H3 and all decision rules, will be registered on the Open Science Framework (OSF) [54] before any model is evaluated on a test fold; because the historical data already exist, the registration will state this, and the hypotheses will additionally be checked on data released after the registration date. The registration identifier will be added to the final version.

VIII. CONCLUSION

This paper presented the design of AI-SMIS, an explainable and evidence-grounded framework for stock-market research on Indian listed companies, together with a complete protocol for testing it. The framework brings together market, financial, text and reference data, applies nine analysis modules and four machine-learning tasks, and presents the results as a score with published weights that can be taken apart exactly, with confidence indicators and a research assistant that answers only from retrieved evidence. The main methodological contributions are a data design that respects the timing of information and the historical composition of the index, a score whose changes can be traced exactly to their causes, and a fully specified evaluation — study period, folds, labels, hyper-parameter ranges, named statistical tests and quantitative decision rules — set out before any results are seen.

Future work has three parts. The first, and most immediate, is to run the Stage-1 experiments in Table 9 on the NIFTY 50, evaluate the research assistant as described in Section V-D, and report all results — including negative ones — against the decision rules stated here, followed by Stage 2 on the NIFTY 500. The second is to extend the system with graph neural networks, larger knowledge graphs, models that read financial documents and tables together, Indian-language news, better portfolio optimisation and real-time monitoring. The third is a user study of how easy to use, clear and useful the platform is. Automated order execution is treated only as a possible future extension and not as part of the system.

ACKNOWLEDGMENT

Dr. Vasudha Patil, the first author, supervised this work as project guide. The authors thank the Department of Artificial Intelligence and Data Science Engineering, Shri Chhatrapati Shivaji Maharaj College of Engineering, Nepti, Ahilyanagar (Maharashtra), for facilities and support during this work.

REFERENCES

[1] S. Gu, B. Kelly, and D. Xiu, "Empirical asset pricing via machine learning," *Rev. Financ. Stud.*, vol. 33, no. 5, pp. 2223–2273, 2020, doi: 10.1093/rfs/hhaa009.

[2] W. Jiang, "Applications of deep learning in stock market prediction: Recent progress," *Expert Syst. Appl.*, vol. 184, Art. no. 115537, 2021, doi: 10.1016/j.eswa.2021.115537.

[3] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, "Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques," *Expert Syst. Appl.*, vol. 42, no. 1, pp. 259–268, 2015, doi: 10.1016/j.eswa.2014.07.040.

© 2026 VT International Press. All rights reserved.

[4] S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman, "Stock price prediction using LSTM, RNN and CNN-sliding window model," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Manipal, India, 2017, pp. 1643–1647, doi: 10.1109/ICACCI.2017.8126078.

[5] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nat. Mach. Intell.*, vol. 1, no. 5, pp. 206–215, 2019, doi: 10.1038/s42256-019-0048-x.

[6] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," arXiv:1702.08608, 2017.

[7] M. López de Prado, *Advances in Financial Machine Learning*. Hoboken, NJ, USA: Wiley, 2018.

[8] D. H. Bailey, J. M. Borwein, M. López de Prado, and Q. J. Zhu, "Pseudo-mathematics and financial charlatanism: The effects of backtest overfitting on out-of-sample performance," *Notices Amer. Math. Soc.*, vol. 61, no. 5, pp. 458–471, 2014.

[9] S. J. Brown, W. N. Goetzmann, R. G. Ibbotson, and S. A. Ross, "Survivorship bias in performance studies," *Rev. Financ. Stud.*, vol. 5, no. 4, pp. 553–580, 1992, doi: 10.1093/rfs/5.4.553.

[10] Z. Ji *et al.*, "Survey of hallucination in natural language generation," *ACM Comput. Surv.*, vol. 55, no. 12, pp. 1–38, 2023, doi: 10.1145/3571730.

[11] P. Islam, A. Kannappan, D. Kiela, R. Qian, N. Scherrer, and B. Vidgen, "FinanceBench: A new benchmark for financial question answering," arXiv:2311.11944, 2023.

[12] H. Kang and X.-Y. Liu, "Deficiency of large language models in finance: An empirical examination of hallucination," in *NeurIPS 2023 Workshop on I Can't Believe It's Not Better (ICBINB)*, 2023, arXiv:2311.15548.

[13] E. F. Fama and K. R. French, "Common risk factors in the returns on stocks and bonds," *J. Financ. Econ.*, vol. 33, no. 1, pp. 3–56, 1993, doi: 10.1016/0304-405X(93)90023-5.

[14] N. Jegadeesh and S. Titman, "Returns to buying winners and selling losers: Implications for stock market efficiency," *J. Finance*, vol. 48, no. 1, pp. 65–91, 1993, doi: 10.1111/j.1540-6261.1993.tb04702.x.

[15] C. R. Harvey, Y. Liu, and H. Zhu, "… and the cross-section of expected returns," *Rev. Financ. Stud.*, vol. 29, no. 1, pp. 5–68, 2016, doi: 10.1093/rfs/hhv059.

[16] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *Eur. J. Oper. Res.*, vol. 270, no. 2, pp. 654–669, 2018, doi: 10.1016/j.ejor.2017.11.054.

[17] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[18] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[19] G. Ke *et al.*, "LightGBM: A highly efficient gradient boosting decision tree," in *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 3146–3154.

[20] P. C. Tetlock, "Giving content to investor sentiment: The role of media in the stock market," *J. Finance*, vol. 62, no. 3, pp. 1139–1168, 2007, doi: 10.1111/j.1540-6261.2007.01232.x.

[21] D. Araci, "FinBERT: Financial sentiment analysis with pre-trained language models," arXiv:1908.10063, 2019.

[22] S. Wu *et al.*, "BloombergGPT: A large language model for finance," arXiv:2303.17564, 2023.

[23] H. Yang, X.-Y. Liu, and C. D. Wang, "FinGPT: Open-source financial large language models," arXiv:2306.06031, 2023.

[24] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 4765–4774.

[25] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, PMLR vol. 70, 2017, pp. 1321–1330.

[26] J. D. Hamilton, "A new approach to the economic analysis of nonstationary time series and the business cycle," *Econometrica*, vol. 57, no. 2, pp. 357–384, 1989, doi: 10.2307/1912559.

[27] A. R. Pagan and K. A. Sossounov, "A simple framework for analysing bull and bear markets," *J. Appl. Econometrics*, vol. 18, no. 1, pp. 23–46, 2003, doi: 10.1002/jae.664.

[28] A. Lunde and A. Timmermann, "Duration dependence in stock prices: An analysis of bull and bear markets," *J. Bus. Econ. Statist.*, vol. 22, no. 3, pp. 253–273, 2004.

[29] P. Lewis *et al.*, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 9459–9474.

[30] S. Yao *et al.*, "ReAct: Synergizing reasoning and acting in language models," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2023, arXiv:2210.03629.

[31] Y. Yu *et al.*, "FinMem: A performance-enhanced LLM trading agent with layered memory and character design," *IEEE Trans. Big Data*, vol. 11, no. 6, 2025 (preprint arXiv:2311.13743).

[32] Y. Xiao, E. Sun, D. Luo, and W. Wang, "TradingAgents: Multi-agents LLM financial trading framework," arXiv:2412.20138, 2024.

[33] Q. Xie *et al.*, "FinBen: A holistic financial benchmark for large language models," in *Adv. Neural Inf. Process. Syst. (Datasets and Benchmarks Track)*, vol. 37, 2024, doi: 10.52202/079017-3033.

[34] S. Es, J. James, L. Espinosa-Anke, and S. Schockaert, "RAGAs: Automated evaluation of retrieval augmented generation," in *Proc. 18th Conf. Eur. Chapter Assoc. Comput. Linguist.: Syst. Demonstrations*, St. Julians, Malta, 2024, pp. 150–158.

[35] R. R. Varghese and B. R. Mohan, "The causal effect of financial news on Indian stock market," in *Proc. IEEE Region 10 Conf. (TENCON)*, 2022, doi: 10.1109/TENCON55691.2022.9977706.

© 2026 VT International Press. All rights reserved.

[36] R. R. Varghese and B. R. Mohan, "Dynamics of nonlinear causality: Exploring the influence of positive and negative financial news on the Indian equity market," in *Proc. Annu. Int. Conf. Emerg. Res. Areas: Int. Conf. Intell. Syst. (AICERA/ICIS)*, IEEE, 2023.

[37] S. A. Balasubramanian, J. N. Christina, and P. Sridevi, "Financial sentiment analysis of news articles with long text corpus for equity portfolio construction," *Int. J. Indian Cult. Bus. Manag.*, vol. 34, no. 3, pp. 389–407, 2025.

[38] Chaithra, K. Kadimisetty, and B. R. Mohan, "Adaptive financial sentiment analysis for NIFTY 50 via instruction-tuned LLMs, RAG and reinforcement learning approaches," arXiv:2512.20082, 2025.

[39] Upstox, "Upstox developer API documentation." [Online]. Available: https://upstox.com/developer/api-documentation/

[40] National Stock Exchange of India, "NSE India." [Online]. Available: https://www.nseindia.com

[41] Securities and Exchange Board of India, "SEBI." [Online]. Available: https://www.sebi.gov.in

[42] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[43] A. Damodaran, *Investment Valuation: Tools and Techniques for Determining the Value of Any Asset*, 3rd ed. Hoboken, NJ, USA: Wiley, 2012.

[44] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. 8th IEEE Int. Conf. Data Min. (ICDM)*, 2008, pp. 413–422, doi: 10.1109/ICDM.2008.17.

[45] A. Grattafiori *et al.*, "The Llama 3 herd of models," arXiv:2407.21783, 2024.

[46] S. Xiao, Z. Liu, P. Zhang, N. Muennighoff, D. Lian, and J.-Y. Nie, "C-Pack: Packed resources for general Chinese embeddings," in *Proc. 47th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2024, doi: 10.1145/3626772.3657878.

[47] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of IR techniques," *ACM Trans. Inf. Syst.*, vol. 20, no. 4, pp. 422–446, 2002.

[48] D. N. Politis and J. P. Romano, "The stationary bootstrap," *J. Amer. Statist. Assoc.*, vol. 89, no. 428, pp. 1303–1313, 1994.

[49] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bull.*, vol. 1, no. 6, pp. 80–83, 1945.

[50] S. Holm, "A simple sequentially rejective multiple test procedure," *Scand. J. Statist.*, vol. 6, no. 2, pp. 65–70, 1979.

[51] Y. Benjamini and Y. Hochberg, "Controlling the false discovery rate: A practical and powerful approach to multiple testing," *J. Roy. Statist. Soc. B*, vol. 57, no. 1, pp. 289–300, 1995.

[52] J. Cohen, "A coefficient of agreement for nominal scales," *Educ. Psychol. Meas.*, vol. 20, no. 1, pp. 37–46, 1960.

[53] Securities and Exchange Board of India, *SEBI (Investment Advisers) Regulations, 2013*. [Online]. Available: https://www.sebi.gov.in/legal/regulations/jan-2013/sebi-investment-advisors-regulations-2013_26872.html

[54] B. A. Nosek, C. R. Ebersole, A. C. DeHaven, and D. T. Mellor, "The preregistration revolution," *Proc. Natl. Acad. Sci. USA*, vol. 115, no. 11, pp. 2600–2606, 2018, doi: 10.1073/pnas.1708274114.

© 2026 VT International Press. All rights reserved.