

Adaptive Position Rebalancing via Regime-Conditioned Contextual Bandits for Pre-Market Trading Optimization

Mutina Sathwik Apuroop

Vellore Institute Of Technology

Avinash Paidi

Amity University

Yamasani Purushotham Reddy

Vellore Institute Of Technology

Harsha Vardhan Akili

Vellore Institute Of Technology

Rolla Subrahmanyam

`rolla.subrahmanyam@vit.ac.in`

Vellore Institute Of Technology

Research Article

Keywords: Contextual bandits, Market-regime, Indian stock market, Pre-Market Trading, Clustering

Posted Date: September 21st, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10882110/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Adaptive Position Rebalancing via Regime-Conditioned Contextual Bandits for Pre-Market Trading Optimization

Mutina Sathwik Apuroop¹, Avinash Paidi^{2*},
Yamasani Purushotham Reddy¹, Harsha Vardhan Akili¹,
Rolla Subrahmanyam^{1*}

¹School of Computer Science and Engineering, Vellore Institute of
Technology Chennai, Chennai, 600127, Tamil Nadu, India.

²Amity School of Engineering and Technology, Amity University,
Hyderabad, 500108, Telangana, India.

*Corresponding author(s). E-mail(s): apaidi@hyd.amity.edu;
rolla.subrahmanyam@vit.ac.in;

Contributing authors: sathwikapuroopmutina@gmail.com;
yamasanipurushothamreddy@gmail.com; akhilharsha78@gmail.com;

Abstract

Managing a portfolio of stocks in a volatile market requires decision making that can adapt to changing market conditions. In the Indian stock market (NSE/BSE), the pre-open session runs from 9:00 AM to 9:08 AM and provides a short time period for adjusting existing positions before regular trading begins. Traditional prediction-based methods may not respond well when market conditions change quickly. In this work, we propose a market-regime-driven contextual bandit framework for selecting daily pre-market portfolio actions: Buy, Sell, or Hold. Instead of using one model for all market conditions, the framework first divides the market into three regimes of Bullish, Bearish, and Neutral using unsupervised clustering of NIFTY 50 market features and technical indicators. For each regime, we maintain an ensemble of four contextual bandit models: Linear Upper Confidence Bound (LinUCB), Linear Thompson Sampling (LinTS), NeuralUCB, and Neural Thompson Sampling (NeuralTS). The final action for each stock is selected through majority voting among the four contextual bandit models associated with the current market regime. The models are updated incrementally as new market observations become available. We evaluate the framework on selected NSE/BSE sectoral stocks and compare its performance with standalone

contextual bandit models as baseline models. The experimental results show that the proposed regime-based approach achieves better performance than the baseline models across the evaluated measures, indicating its potential as an adaptive decision-support approach for pre-market portfolio management.

Keywords: Contextual bandits, Market-regime, Indian stock market, Pre-Market Trading, Clustering

1 Introduction

Decision-making in stock markets is difficult because financial data are highly volatile, non-stationary, and often non-linear. Market conditions can change because of changes in liquidity, volatility, investor sentiment, and broader economic factors. Such changes are particularly important for short-term portfolio management, where traders need to make position adjustments using the latest available market information.

In the Indian stock market (NSE/BSE), the daily pre-open session runs from 9:00 AM to 9:08 AM and provides a short period in which traders can assess the market before regular trading begins. For a trader who already holds a portfolio of stocks, this creates a simple but important decision: whether to increase the position (Buy), reduce or close the position (Sell), or keep the current position unchanged (Hold) for the upcoming trading session. In this work, we formulate this problem as a sequential next-day pre-open portfolio decision-making task. Rather than predicting only the future direction of stock prices, the proposed framework uses information from the previous trading day to directly select an appropriate portfolio action. This formulation is consistent with computational economics, where data-driven and simulation-based approaches are increasingly used to address complex financial decision-making problems [1–4].

Traditional machine learning models have shown good performance in learning non-linear patterns for stock price prediction and market anomaly detection [5–9]. However, these models are generally designed as static predictors. In portfolio management, the decision process is different because market conditions change continuously and the available information is updated over time. Therefore, traders need an online decision-making system that can use the current market context to select an action and update its decisions as new observations become available.

Contextual bandits are well suited to this type of sequential decision-making problem. In a contextual bandit setting, an agent observes the current context, selects an action, receives a reward, and then uses the observed feedback to update its policy. Linear contextual bandit methods, such as Linear Upper Confidence Bound (LinUCB) and Linear Thompson Sampling (LinTS), are computationally efficient and can be updated online as new observations become available [10, 11]. Neural versions, such as NeuralUCB and NeuralTS, use neural networks to model more complex and non-linear relationships between the context and the reward [12, 13].

However, the non-stationary nature of financial markets creates a challenge for standard contextual bandit methods. A policy that performs well during a stable

upward market may not perform equally well during a sudden market decline or a period of high volatility. Regime-switching models have been used to capture structural changes in financial time series [14, 15], while non-stationary bandit methods have been developed to improve adaptation to changing environments [16, 17]. However, the use of market-regime clustering together with an ensemble of contextual bandits for next-day pre-market trading decisions has received limited attention.

To address this problem, we propose a market-regime-driven contextual bandit framework for next-day pre-open portfolio decision-making. Instead of using a single model for all market conditions, the framework first identifies the current market regime using unsupervised clustering of NIFTY 50 features, including returns, volatility, and technical indicators. The market is divided into three regimes: Bullish, Bearish, and Neutral. A separate ensemble of four contextual bandit algorithms (LinUCB, NeuralUCB, LinTS, and NeuralTS) is maintained for each regime. For a stock in the portfolio, the final trading action is selected using majority voting among the four bandit models associated with the current market regime. The models are then updated as new market observations become available, allowing the framework to adapt to changes in market conditions.

We evaluate the proposed framework using historical data from selected Indian sectoral stocks and the NIFTY 50 index covering the period from January 2015 to July 2025. The out-of-sample evaluation covers the period from July 2024 to July 2025. The results show that the proposed regime-conditioned framework performs better than the standalone contextual bandit baselines in terms of cumulative reward, percentage of optimal actions, and cumulative regret.

The remainder of this paper is organized as follows. Section 2 reviews related work in computational finance, contextual bandits, and market-regime modelling. Section 3 introduces the mathematical notation and problem formulation. Section 4 describes the proposed framework in detail. Section 5 presents the experimental setup and evaluation results, and Section 6 concludes the paper and discusses possible directions for future work.

2 Related Work

Selecting the appropriate trading action (Buy, Sell, or Hold) for a portfolio of stocks before the next trading day requires a method that can respond to changing market conditions. Previous research related to this problem can mainly be grouped into three areas: contextual bandits, market regime modelling, and algorithmic trading. Each of these areas provides useful ideas for the proposed framework, but there is still a lack of an integrated approach specifically designed for next-day pre-market decision support. This section reviews the relevant work and discusses the gaps addressed by our study.

2.1 Contextual Bandits for Financial Decision-Making

Contextual multi-armed bandits provide a useful framework for making sequential decisions when the outcome of an action is uncertain. At each step, the agent observes a context, selects an action, receives a reward, and uses this feedback to improve future decisions. Linear methods such as LinUCB [10] and Linear Thompson Sampling

(LinTS) [11] are widely used because they are computationally efficient and can be updated as new observations become available. However, these methods assume a linear relationship between the context and the expected reward, which may not be sufficient for financial data with complex and non-linear patterns.

NeuralUCB [12] and Neural Thompson Sampling (NeuralTS) [13] address this issue by using neural networks to model non-linear relationships. These methods have also been explored in areas such as dynamic portfolio management [18] and derivatives pricing [19]. However, existing applications generally focus on predicting continuous rewards or making decisions for individual assets. The problem of selecting discrete Buy, Sell, or Hold actions for several stocks within a portfolio has received less attention. In addition, many existing financial bandit approaches are evaluated under relatively stable market assumptions. Some studies have considered regime information to deal with non-stationarity, such as the approach in [20], but these methods have not been specifically developed for next-day pre-market decision-making.

2.2 Market Regime Modelling and Clustering Approaches

Financial markets do not remain in the same state over time. They can move between Bullish, Bearish, and Neutral conditions as volatility, momentum, liquidity, and other market factors change. Hidden Markov Models (HMMs) have traditionally been used to identify such changes in financial time series [14, 15]. Unsupervised learning methods provide another way to identify market regimes without requiring predefined state labels. For example, K-Means clustering has been used to group market observations based on volatility measures and technical indicators [21]. Other commonly used approaches include hierarchical clustering [22], spectral clustering [23], and self-organizing maps [24].

Market regime information has been useful in applications such as volatility timing [25], momentum analysis, and asset allocation. However, most of these approaches use the identified regimes to improve forecasts or determine broader investment strategies rather than to directly select discrete trading actions for individual stocks. In particular, the combination of unsupervised market regime identification with contextual bandit models, where the selected model is updated as market conditions change, has received limited attention. This gap motivates the regime-based contextual bandit framework proposed in this work.

2.3 Algorithmic Trading Systems and Pre-Market Decision Frameworks

Algorithmic trading research includes a wide range of supervised learning and reinforcement learning methods. Deep learning models such as Long Short-Term Memory (LSTM) networks [26] and Convolutional Neural Networks (CNNs) [27] have been used to learn temporal patterns from financial data. However, these models generally produce price or return estimates, which then need to be converted into trading decisions using an additional rule or processing step. Training such models can also require considerable computational resources, particularly when several stocks need to be updated regularly. Other studies have used additional information sources, such as

microblogging sentiment, to improve stock prediction through hybrid neural network models [28]. Although these approaches can provide useful predictive information, they do not directly address the online action-selection problem considered in this work.

Reinforcement learning methods such as Deep Q-Networks (DQN) [29], Proximal Policy Optimization (PPO) [30], and actor-critic methods [31] have also been applied to algorithmic trading. These methods learn policies by considering sequences of states and actions over longer investment periods. However, they can require a large amount of data and computational resources, and updating the policy frequently can be difficult. They are also more complex than necessary for a decision that only concerns the next trading day. The contextual bandit formulation used in our work avoids the need to model long sequences of future states and instead focuses on selecting the best immediate action from the current market context.

The pre-market period has received comparatively little attention in quantitative trading research. Most existing studies focus on intraday trading or end-of-day decisions. Although pre-market activity has been studied in some international markets, the short call-auction mechanism used during the Indian NSE/BSE pre-open session, from 9:00 AM to 9:08 AM, has received limited attention from adaptive learning approaches. This creates an opportunity to study pre-market trading as a short-horizon action-selection problem.

2.4 Summary of Gaps and Our Contribution

The review of existing work points to four main gaps:

1. Contextual bandit methods have not been widely studied for generating next-day Buy/Sell/Hold recommendations for individual stocks already held in a portfolio.
2. Market regime identification has not been combined with separate contextual bandit ensembles that are selected according to the current regime.
3. Many financial learning approaches do not explicitly address continuous daily adaptation to changing market conditions through incremental model updates.
4. The short pre-open call-auction period in the Indian stock market has received limited attention as a discrete action-selection problem.

Our work addresses these gaps by combining linear and neural contextual bandit models with unsupervised market regime clustering and incremental daily updates. The resulting framework is designed to provide direct Buy, Sell, or Hold recommendations for stocks in a portfolio before the Indian pre-market session begins.

3 Preliminaries and Problem Formulation

This section introduces the mathematical notation used in the proposed framework. It first defines the available trading actions and then describes the technical indicators used to construct the context vector. The contextual bandit models are introduced in the following subsections.

3.1 Trading Action Space

For each stock in the portfolio, the agent can choose one of three possible actions:

$$\mathcal{A} = \text{Buy, Sell, Hold.} \quad (1)$$

The **Buy** action means increasing the current position in a stock when the model expects a positive price movement. The **Sell** action means reducing or closing the existing position when a negative price movement is expected. The **Hold** action leaves the current position unchanged when there is no sufficiently strong signal to buy or sell. The action is selected before the next trading session using the market information available up to that point.

3.2 Technical Context Indicators

The context vector x_t contains information about recent price and trading-volume behavior of a stock on day t . We use a set of commonly used technical indicators to describe the recent market condition. The indicators used in the framework are defined below:

- **Daily Returns:** The daily return represents the relative change in the closing price from one trading day to the next:

$$r_t = \frac{P_t - P_{t-1}}{P_{t-1}}, \quad (2)$$

where P_t is the closing price on day t .

- **Simple Moving Average (SMA):** The SMA represents the average closing price over the previous k trading days. It is calculated as:

$$\text{SMA}_k(t) = \frac{1}{k} \sum_{i=0}^{k-1} P_{t-i}. \quad (3)$$

We use two moving averages, namely the 5-day SMA (SMA_5) and the 20-day SMA (SMA_{20}), to represent short- and medium-term price trends.

- **Exponential Moving Average (EMA):** The EMA gives more weight to recent prices and therefore responds faster to price changes than the SMA. It is calculated recursively as:

$$\text{EMA}_k(t) = \alpha_k P_t + (1 - \alpha_k) \text{EMA}_k(t-1), \quad (4)$$

where $\alpha_k = \frac{2}{k+1}$.

- **Moving Average Convergence Divergence (MACD):** MACD is used to represent price momentum and is calculated from the difference between the 12-day and 26-day EMAs:

$$\text{MACD}_t = \text{EMA}_{12}(t) - \text{EMA}_{26}(t). \quad (5)$$

- **Relative Strength Index (RSI):** RSI is a momentum indicator that measures the relative strength of recent price gains and losses. For a 14-day period, it is

calculated as:

$$RSI_t = 100 - \left(\frac{100}{1 + RS_t} \right), \quad (6)$$

where RS_t is the ratio of the average gain to the average loss over the previous 14 trading days.

- **On-Balance Volume (OBV):** OBV combines price direction with trading volume and is used to represent changes in buying and selling activity:

$$OBV_t = OBV_{t-1} + \text{sgn}(P_t - P_{t-1}) \cdot V_t, \quad (7)$$

where V_t represents the trading volume on day t , and $\text{sgn}(\cdot)$ denotes the sign function.

- **Price Momentum:** Price momentum measures the relative price change over the previous five trading days:

$$\text{Momentum}_t = \frac{P_t - P_{t-5}}{P_{t-5}}. \quad (8)$$

- **Trend Indicator:** This binary feature indicates whether the current price is above the 20-day moving average:

$$\text{Trend}_t = \begin{cases} 1, & \text{if } P_t > \text{SMA}_{20}(t), \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

3.3 Market Regime Identification via Clustering

To account for changes in market conditions, we divide the daily market environment into three regimes: Bullish, Bearish, and Neutral. The regimes are identified using K -Means clustering on features obtained from the NIFTY 50 index. The clustering objective is to minimize the within-cluster variance:

$$J = \sum_{i=1}^n \sum_{k=1}^K w_{ik} |x_i - \mu_k|^2, \quad (10)$$

where $w_{ik} = 1$ when the data point x_i belongs to regime k and $w_{ik} = 0$ otherwise. The term μ_k denotes the centroid of the k -th regime. The resulting clusters are used to determine the current market regime before selecting the trading action.

3.4 Contextual Bandit Foundations

We use four contextual bandit algorithms as the decision-making models: LinUCB, LinTS, NeuralUCB, and NeuralTS. At each time step, these models receive the available technical features as context and select one of the possible trading actions based on the estimated reward.

3.4.1 Linear Contextual Bandits

- **LinUCB**: LinUCB assumes that the expected reward is a linear function of the context features. Let $\theta \in \mathbb{R}^d$ denote the unknown model parameter. The expected reward is given by:

$$\mathbb{E}[r_t | x_{t,a}] = \theta^T x_{t,a}. \quad (11)$$

To balance exploration and exploitation, LinUCB selects the action with the highest upper confidence bound:

$$a_t = \arg \max_{a \in \mathcal{A}} \left(\theta^T x_{t,a} + \alpha \sqrt{x_{t,a}^T A^{-1} x_{t,a}} \right), \quad (12)$$

where A is the context covariance matrix and $\alpha > 0$ controls the level of exploration.

- **Linear Thompson Sampling (LinTS)**: LinTS uses a probability distribution over the model parameters instead of directly using a single parameter estimate. At time t , the parameter is sampled from the posterior distribution:

$$\theta_t \sim \mathcal{N}(\mu_t, \Sigma_t). \quad (13)$$

The selected action is the one with the highest estimated reward under the sampled parameter:

$$a_t = \arg \max_{a \in \mathcal{A}} \theta_t^T x_{t,a}. \quad (14)$$

3.4.2 Neural Contextual Bandits

Linear models may not be sufficient when the relationship between the context and reward is highly non-linear. Neural contextual bandit methods address this issue by using a neural network $f_\theta(x_{t,a})$ to estimate the expected reward.

- **NeuralUCB**: NeuralUCB uses a neural network to estimate the reward for each action. The network parameters θ are updated using gradient-based optimization. The action is selected using a confidence-based rule:

$$a_t = \arg \max_{a \in \mathcal{A}} (f_\theta(x_{t,a}) + \alpha \sigma_{t-1}(x_{t,a})), \quad (15)$$

where $\sigma_{t-1}(x_{t,a})$ represents the estimated uncertainty for the given context based on the observations collected so far.

- **Neural Thompson Sampling (NeuralTS)**: NeuralTS combines neural networks with Thompson Sampling to introduce exploration. At each step, network parameters are sampled from a Gaussian distribution around the current parameter estimate:

$$\theta_t \sim \mathcal{N}(\mu_\theta, \sigma_\theta^2 I). \quad (16)$$

The sampled network is then used to estimate the reward for each available action:

$$a_t = \arg \max_{a \in \mathcal{A}} f_{\theta_t}(x_{t,a}). \quad (17)$$

3.5 Ensemble Decision Mapping

The final trading decision is obtained by combining the predictions from the four contextual bandit models. For the current market regime k , each localized bandit model independently selects one of the available actions. The four recommendations are then combined using majority voting, and the action with the most votes is selected as the final decision.

If the four models produce a tie, a regime-specific preference rule is used to select the final action. For example, the framework can prefer ‘Buy’ in a Bullish regime and ‘Hold’ in a Neutral regime. This ensemble approach allows the framework to use the strengths of different bandit models while adapting the decision process according to the current market regime.

4 Proposed Methodology

This section describes the technical architecture of the proposed market regime-driven contextual bandit framework. The overall workflow is shown in Figure 1. The framework follows a day-by-day learning process in which new market data are processed, the current market regime is identified, and the bandit models are updated based on the latest observations. This setup is intended to reflect how the framework can be used in an actual trading environment.

The proposed framework follows a sequence of steps that starts with collecting the financial data and ends with updating the trading models. The main steps include data preprocessing, feature extraction, market regime identification, selection of the corresponding bandit models, and model updates. Instead of using the same model for all market conditions, the framework first identifies the broader market regime using NIFTY 50 features and then uses the models associated with that regime.

4.1 Data Preprocessing and Feature Extraction

The process starts with collecting daily price and trading volume data for the stocks in the portfolio and the NIFTY 50 index. A rolling lookback window of w days is used to include recent observations when constructing the input features.

The raw market data are then converted into a context vector x_t . This vector contains information about recent price movements, returns, trading volume, and market trends. The features include indicators such as RSI, MACD, OBV, and SMAs. These context vectors are provided to the contextual bandit models when selecting the trading action.

4.2 Market State Clustering and Reward Formulation

The framework divides the market into three regimes: Bullish, Bearish, and Neutral. These regimes are identified using K -Means clustering on features calculated from the NIFTY 50 index. The clustering process groups similar market conditions and assigns the current observation to a market state $c_t \in 1, 2, 3$. The identified state is then used to select the corresponding set of bandit models.

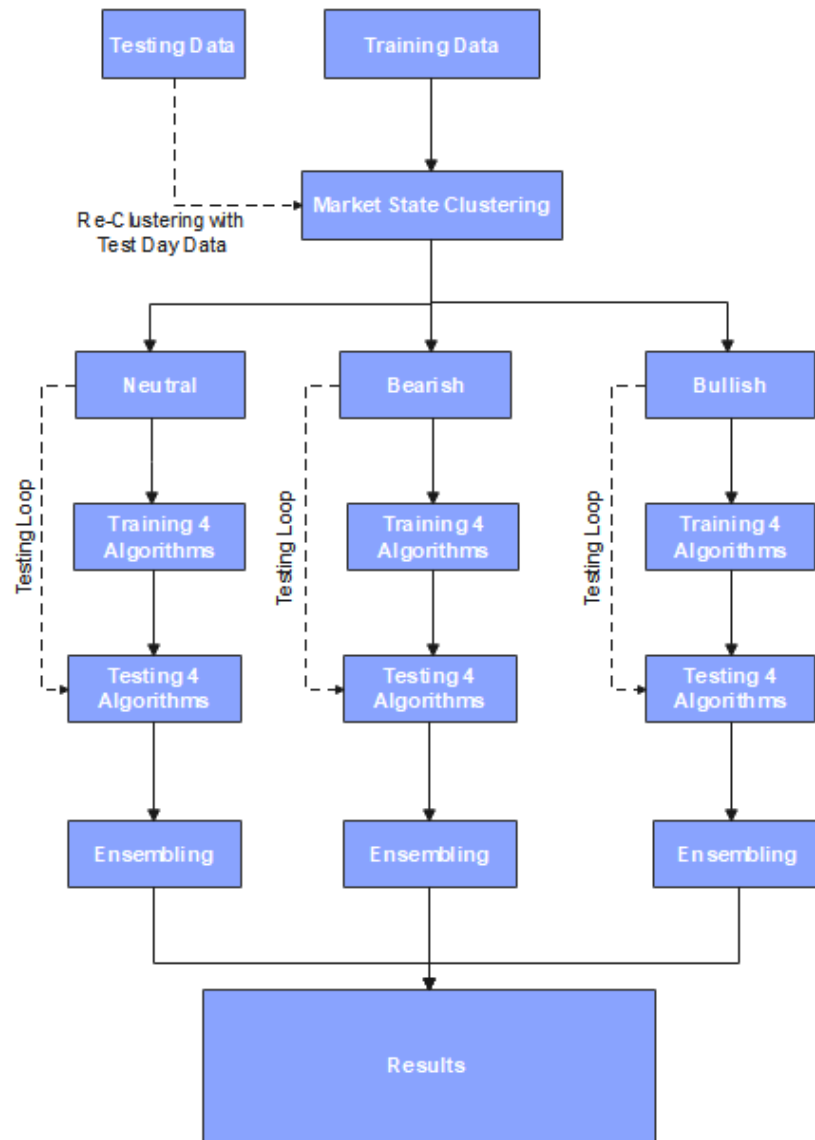


Fig. 1 Workflow of the Proposed Market Regime-Driven Framework

The reward is calculated using a dynamic threshold rather than a fixed percentage change. For each trading day, the actual change in the stock price is compared with a threshold τ_t . The threshold is calculated using the average absolute price change observed up to day t :

$$\tau_t = \frac{1}{t} \sum_{i=1}^t |P_i - P_{i-1}|. \quad (18)$$

When the price change remains within the range $[-\tau_t, \tau_t]$, the movement is treated as relatively small and the **Hold** action receives the highest reward. A price increase greater than τ_t results in a positive reward for the **Buy** action, while a decrease below $-\tau_t$ results in a positive reward for the **Sell** action. The threshold changes with the observed price movements, allowing the reward definition to adjust to different levels of market volatility.

4.3 Model Training Phase

The training phase, shown in Algorithm 1, creates three specialist modules corresponding to the three market regimes. The models are trained using expanding historical data blocks so that the training process follows the same sequential pattern as an online trading setting.

For each data block, the system first identifies the current market regime c_t using the NIFTY 50 index features. The stock context vector x_t is then sent to the specialist model m_{c_t} associated with that regime. Each specialist consists of four independent contextual bandit models: LinUCB, LinTS, NeuralUCB, and NeuralTS. Each model selects an action and receives a reward based on the stock’s price movement on the following trading day. After the data block is processed, the parameters of the active models are updated using the newly observed information. Over time, this allows each specialist to learn from the market conditions associated with its regime.

4.4 Streaming Testing and Real-Time Prediction

In an actual deployment, the model would process each new trading day without knowing the market information that will become available in the future. To reproduce this setting, the testing phase in Algorithm 2 is carried out as a strict day-by-day streaming simulation.

For each new trading day, the framework first constructs the stock context vector using the latest available price and technical indicator data. It then identifies the current market regime (*Bullish*, *Bearish*, or *Neutral*) using the corresponding NIFTY 50 index features. The specialist model associated with the identified regime is then activated. Its four bandit models, namely LinUCB, LinTS, NeuralUCB, and NeuralTS, independently select an action, and a majority vote is used to determine the final **Buy**, **Sell**, or **Hold** decision.

After the next trading day ends, the actual price movement becomes available, and the corresponding environmental reward is calculated. This reward is then used to update the parameters of the active specialist model. To account for longer-term changes in market behavior, the clustering centroids are also updated periodically.

Algorithm 1 Model Training Phase

```
1: Input: Stock data  $D$ , market index data  $I$ , portfolio stock set  $S$ , lookback window  $w$ , base batch size  $b_{\text{base}}$ , cluster count  $k = 3$ 
2: Output: Trained regime-specific specialist models  $M = \{m_1, m_2, m_3\}$ 
3: Initialize specialized bandit models  $m_c$  for each regime  $c \in \{1, \dots, k\}$ 
4: Initialize empty historical memories  $\mathcal{D}_c \leftarrow \emptyset$ 
5: Extract price, volume, and technical indicators for each trading day  $t$ 
6: Construct context vectors  $x_t$  using lookback window  $w$ 
7: Train initial  $K$ -Means centroids on historical index features to define assignment function  $\text{cluster}(x)$ 
8: Partition full training dataset  $D$  into sequential batches  $B = \{D^{(1)}, D^{(2)}, \dots, D^{(n)}\}$  of size  $b_{\text{base}}$ 
9: for each batch  $D^{(i)} \in B$  do
10:   for each streaming day  $t$  within batch  $D^{(i)}$  do
11:     Identify prevailing market regime:  $c_t \leftarrow \text{cluster}(I_t)$ 
12:     Route sample to relevant memory:  $\mathcal{D}_{c_t} \leftarrow \mathcal{D}_{c_t} \cup \{x_t\}$ 
13:     Activate corresponding specialized model  $m_{c_t}$ 
14:     Select action  $a_t = m_{c_t}.\text{select\_action}(x_t)$  via internal ensemble vote
15:     Observe next-day market response and receive reward  $r_t$ 
16:     Update model parameters  $m_{c_t}$  incrementally using tuple  $(x_t, a_t, r_t)$ 
17:   end for
18:   for each regime channel  $c \in \{1, 2, 3\}$  do
19:     if  $|\mathcal{D}_c| > 0$  then
20:       Fine-tune specialist weights  $m_c$  on its accumulated history  $\mathcal{D}_c$ 
21:     end if
22:   end for
23: end for
24: return  $M$ 
```

using the additional historical data. In this way, both the market regime representation and the bandit models can gradually adjust as new market information becomes available.

5 Experimental Setup and Results

This section presents the experimental evaluation of the proposed framework. It describes the dataset, preprocessing steps, feature construction, model settings, and the comparison with baseline contextual bandit models across different stock sectors.

5.1 Experimental Setup

5.1.1 Data Collection and Source Pipeline

We collected historical daily market data for selected Indian sectoral stocks and the NIFTY 50 index using the yfinance API. The data cover the period from January 2015 to July 2025 and include daily opening, high, low, and closing prices, along

Algorithm 2 Streaming Evaluation and Live Update Phase

```
1: Input: Evaluation stream  $D_{\text{test}}$ , active index data  $I_{\text{test}}$ , specialists  $M = \{m_1, m_2, m_3\}$ , lookback window  $w$ 
2: Output: Evaluation performance metrics Results
3: Initialize running performance monitor Results  $\leftarrow \emptyset$ 
4: for each incoming trading day  $t \in D_{\text{test}}$  do
5:   Build context vector  $x_t$  using the previous  $w$  days of market data
6:   Determine current market regime using active index features:  $c_t \leftarrow \text{cluster}(I_t)$ 
7:   Retrieve the designated specialist model  $m_{c_t}$ 
8:   Generate the portfolio action recommendation:  $a_t \leftarrow m_{c_t}.\text{select\_action}(x_t)$ 
   // Market executes trade and closes
9:   Observe next-day closing price  $P_{t+1}$  and calculate dynamic threshold  $\tau_{t+1}$ 
10:  Compute realized environmental reward  $r_t$  based on action  $a_t$  and price change
11:  Update the active specialist parameters  $m_{c_t}$  incrementally with  $(x_t, a_t, r_t)$ 
12:  Log performance metrics and realized profits into Results
13:  Periodically update  $K$ -Means cluster centroids using all index data observed up to time  $t$ 
14: end for
15: return Results
```

with trading volume. To include stocks from different sectors, we selected Hindustan Unilever (FMCG), HDFC Bank (Banking), Infosys (Information Technology), Larsen & Toubro (Infrastructure), and Mahindra & Mahindra (Automotive). The selected period includes different market conditions and economic cycles, which allows the framework to be evaluated over a relatively wide range of market situations.

5.1.2 Data Preprocessing

The collected data were preprocessed before being used for training and evaluation. The following steps were applied:

- **Missing Value Imputation:** Missing observations caused by market holidays and other non-trading days were handled using forward filling followed by backward filling. This was done to maintain continuity in the time-series data.
- **Feature Normalization:** The features have different numerical ranges, particularly price-based variables and technical indicators. Therefore, continuous features were standardized using:

$$x' = \frac{x - \mu}{\sigma}, \quad (19)$$

where μ and σ represent the mean and standard deviation, respectively. These values were calculated only from the training data to prevent information from the test period from being used during training.

- **Outlier Preservation:** Large price movements and periods of high volatility were retained in the dataset. Removing these observations could make the evaluation less representative of actual market conditions, where sudden price changes can occur.

5.1.3 Feature and Context Formulation

For each stock, we used a rolling window of $w = 10$ trading days to construct the daily context vector x_t . The purpose of this window is to provide the model with recent information about price movements, trends, and trading activity. The context vector contains daily log-returns, short- and medium-term moving averages (SMA * 5 and SMA * 20), MACD values, relative strength index (RSI₁₄), on-balance volume (OBV_t), and a binary indicator representing trend persistence.

5.1.4 Market State Clustering Parameters

Market regimes were identified using technical features calculated from the NIFTY 50 index. Ten features were used for this purpose: daily returns, RSI, MACD, historical volatility, intraday returns, overnight returns, the close-to-SMA ratio, OBV velocity, Average Directional Index (ADX), and the trailing 20-day SMA. We applied K -Means clustering with $K = 3$ to divide the observations into three market regimes: Bullish, Neutral, and Bearish. The clustering objective is given by:

$$J = \sum_{k=1}^3 \sum_{x_i \in C_k} |x_i - \mu_k|^2, \quad (20)$$

where C_k denotes the k -th market regime and μ_k is the centroid of that cluster. During the online evaluation, the cluster representation is updated as new market observations become available.

5.1.5 Training Protocol and Baselines

The complete dataset was divided into two periods. Data from January 2015 to June 2024 were used for training, while data from July 2024 to July 2025 were used for out-of-sample streaming evaluation. At each time step, the reward r_t was determined by comparing the selected action with the next-day price movement:

$$r_t = \begin{cases} +1, & \text{if } a_t = \text{Buy and } P_{t+1} > P_t + \tau_t, \\ +1, & \text{if } a_t = \text{Sell and } P_{t+1} < P_t - \tau_t, \\ -1, & \text{if } a_t = \text{Hold and } |P_{t+1} - P_t| \leq \tau_t, \\ \text{otherwise,} & \end{cases} \quad (21)$$

where τ_t denotes the dynamic volatility threshold.

For comparison, we used four contextual bandit models: LinUCB, LinTS, NeuralUCB, and NeuralTS. Each baseline used a single global market representation and did not use the proposed market-regime clustering. The same hyperparameter settings were used for both the baseline models and the proposed framework to maintain a consistent comparison. The parameter settings are reported in Table 1.

5.2 Empirical Results and Analysis

The out-of-sample results from the streaming test period show that the proposed regime-aware framework performs better than the baseline models that do not consider

Table 1 Hyperparameter Configuration Framework. This table details the core hyperparameter selections and architecture settings applied to both the standalone baseline benchmarks and the regime-specific specialist models.

Model Level	Key Parameter	Value
LinUCB	Exploration Parameter (α)	0.300
LinTS	Prior Scaling Variance (σ)	0.100
NeuralUCB	Hidden LSTM Architecture Units	128
NeuralTS	Optimization Learning Rate	0.001
K-Means Clustering	Total Target Regimes (K)	3

market regimes. Table 2 reports the trading decision accuracy obtained for the stocks from different sectors.

Table 2 Trading Decision Accuracy Percentage Comparison. This table presents the percentage of trading days where the selected portfolio action matches the tracking market direction during the out-of-sample evaluation window (July 2024 to July 2025).

Stock Ticker	LinUCB	LinTS	NeuralUCB	NeuralTS	Proposed Framework
HINDUNILVR.NS	44.00	42.40	46.00	34.80	61.20
HDFCBANK.NS	52.00	50.80	48.80	36.80	60.80
INFY.NS	49.60	51.60	49.60	36.40	57.20
LT.NS	48.00	47.20	40.00	35.60	56.00
M&M.NS	46.80	47.20	48.00	34.80	54.80

The results show that the proposed framework performs better than the standalone baseline models across the evaluated stocks. By assigning each context to a model associated with the current market regime, the framework can adapt its decisions to changing market conditions. This helps reduce the policy drift observed in the baseline models when the market becomes non-stationary.

The deep-learning-based models, NeuralUCB and NeuralTS, perform better than the linear models for some individual stocks. However, when used as global models, they also face difficulties when market conditions change suddenly. The proposed regime-based approach shows a more consistent increase in cumulative reward over the evaluation period, as shown in Figure 2. These results suggest that using separate policies for different market regimes can improve the consistency of the trading decisions and lead to more stable rewards.

The difference in performance can also be seen in the percentage of optimal actions selected over time, as shown in Figure 3. After the initial exploration period, the proposed framework maintains a higher proportion of optimal actions than the baseline models. This suggests that identifying different market regimes helps the bandit agents work with a more specific set of market conditions. As a result, the agents can learn

Model Performance Comparison: Cumulative Reward

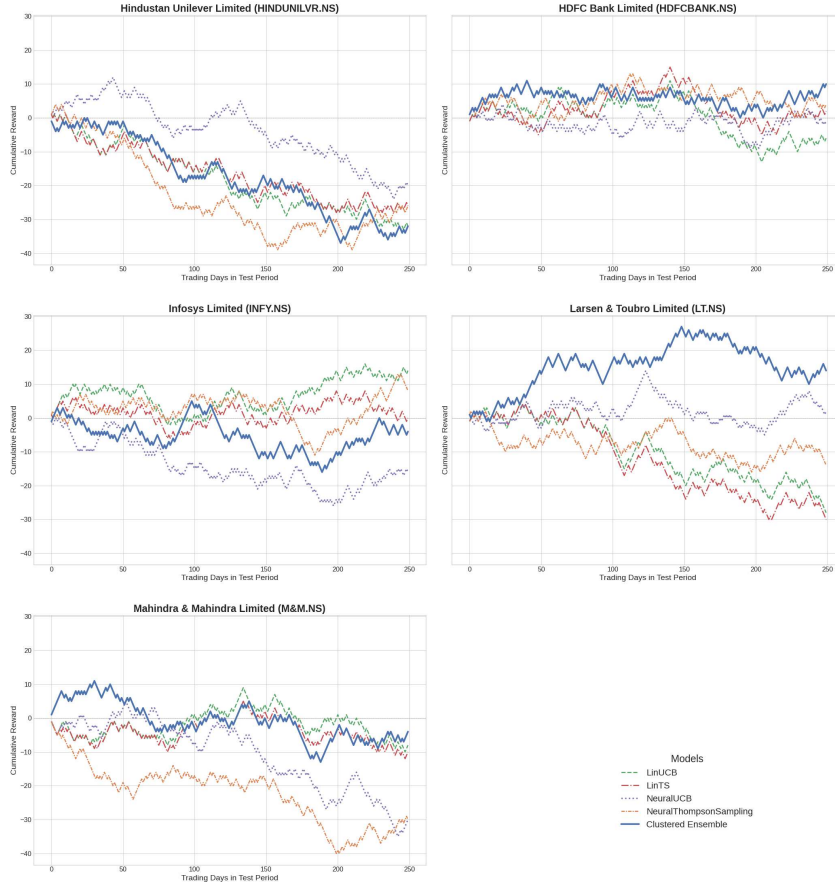


Fig. 2 Out-of-Sample Cumulative Reward Progression Tracking Path

the appropriate actions more quickly and achieve a better balance between exploration and exploitation.

Finally, Figure 4 shows the cumulative tracking regret over the test period. Tracking regret represents the difference between the actions selected by the model and those selected by an oracle policy that always chooses the best possible action. The proposed framework accumulates regret at a lower rate across the evaluated stocks. This result indicates that the framework can maintain more consistent decisions under changing and volatile market conditions, with less degradation in performance over time.

Model Performance Comparison: % of Optimal Action Taken

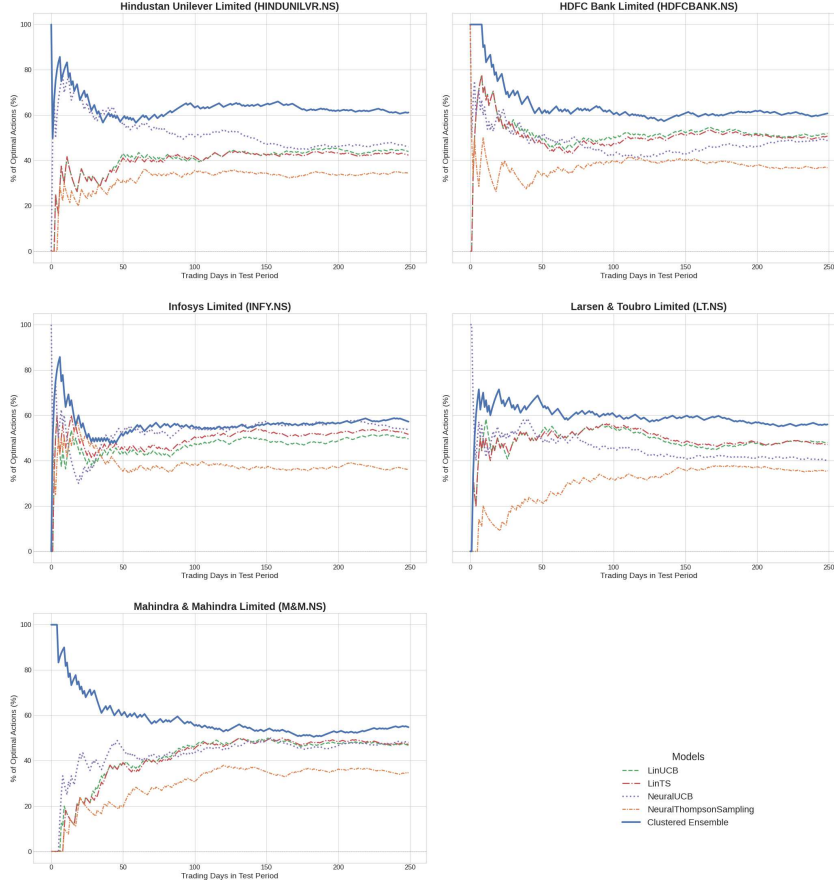


Fig. 3 Percentage of Optimal Asset Allocation Actions Selected Over Time

6 Conclusion, Limitations, and Future Work

This section discusses the key findings of this paper, outlines the limitations of the proposed framework, and identifies potential future directions for extending the adaptive pre-market decision support framework.

6.1 Conclusion and Core Findings

This paper presents a market-regime-based contextual bandit framework for helping traders decide how to adjust their stock positions before the next trading day. The framework considers three possible actions: Buy, Sell, and Hold. It first identifies broad market conditions through unsupervised clustering of macroeconomic indicators and then selects actions using a localized ensemble of linear and neural contextual bandit methods, including LinUCB, LinTS, NeuralUCB, and NeuralTS. Unlike conventional

Model Performance Comparison: Cumulative Regret

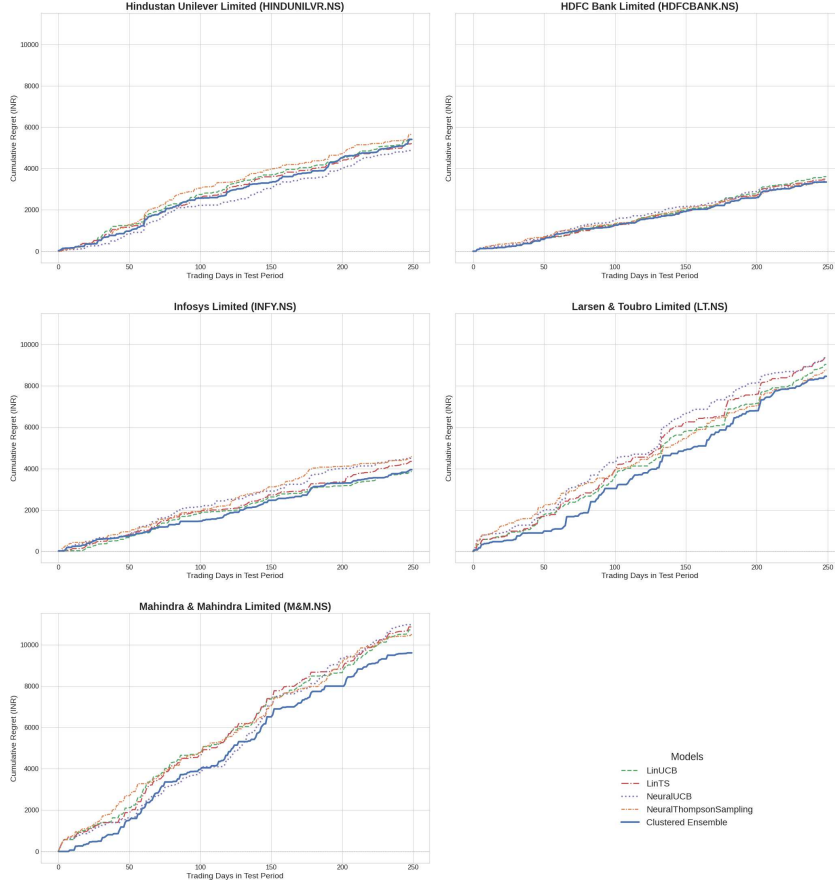


Fig. 4 Cumulative Regret Path Analysis Against Baseline Operations

approaches that rely on fixed forecasting models, the proposed framework updates its decisions as market conditions change. The model is evaluated using a day-by-day streaming simulation, where market regimes and model parameters are updated sequentially to reflect the way the framework would operate in a live trading setting.

Our experiments on major Indian sectoral stocks show that financial non-stationarity can significantly affect the performance of a single global bandit model. The proposed regime-conditioned approach, in comparison, achieves lower cumulative tracking regret and higher cumulative rewards across the evaluated settings. A key reason for this improvement is the use of separate models for Bullish, Bearish, and Neutral market conditions. By learning within a more specific market context, each model deals with a relatively simpler reward structure and can adapt more effectively to changes in the environment. The results therefore indicate that conditioning the decision process on market regimes can improve the stability and effectiveness of next-day position adjustments, particularly under volatile market conditions.

6.2 System Boundaries and Limitations

Although the proposed framework provides a strong benchmark, its current deployment is subject to two important limitations:

- **Information Asymmetry:** The context vector is derived entirely from historical endogenous market information, including price movements, trading volume, and trailing technical indicators. Therefore, the framework has limited ability to account for unexpected external events, such as overnight changes in global economic policies, corporate earnings announcements, or sudden geopolitical developments. Such events can lead to substantial price gaps at the opening of the market.
- **Frictionless Execution:** The current environment calculates rewards under idealized trading conditions and does not explicitly model practical market frictions. Factors such as brokerage charges, exchange transaction taxes, liquidity constraints, and execution slippage are not included in the simulation. These factors may have a noticeable effect on actual performance, particularly during the short 8-minute pre-open call-auction period.

6.3 Future Research Directions

There are several ways in which the proposed pre-market trading framework can be extended in future work:

- **Multimodal Context Fusion:** The current framework mainly uses market data from previous trading days. In future, we can also include information from financial news and social media. These sources may provide useful information about changes in market sentiment, especially when important events occur outside regular trading hours.
- **Probabilistic State Spaces:** The current framework uses clustering to identify different market states. Future work can explore probabilistic models such as Hidden Markov Models (HMMs) and deep temporal clustering methods. These approaches may help in identifying gradual or uncertain changes between market regimes.
- **Microstructure and Friction Constraints:** Future experiments can include order-book information, such as bid–ask spreads and order imbalance, along with the existing market features. The reward calculation can also include brokerage charges, transaction costs, and other execution costs. This would make the experimental setting closer to actual trading conditions.

References

- [1] Amman, H.M.: What is computational economics? *Computational Economics* **10**(2), 103–105 (1997) <https://doi.org/10.1023/A:1008661319094>
- [2] Tesfatsion, L., Judd, K.L. (eds.): *Handbook of Computational Economics, Volume 2: Agent-Based Computational Economics*. Elsevier, ??? (2006)
- [3] Gogas, P., Papadimitriou, T.: Machine learning in economics and finance. *Computational Economics* **57**(1), 1–4 (2021) <https://doi.org/10.1007/>

- [4] Charpentier, A., Élie, R., Remlinger, C.: Reinforcement learning in economics and finance. *Computational Economics* **62**(1), 425–462 (2023) <https://doi.org/10.1007/s10614-021-10119-4>
- [5] Gu, S., Kelly, B., Xiu, D.: Empirical asset pricing via machine learning. *The Review of Financial Studies* **33**(5), 2223–2273 (2020) <https://doi.org/10.1093/rfs/hhaa009>
- [6] Zhang, J., Li, L., Chen, W.: Predicting stock price using two-stage machine learning techniques. *Computational Economics* **57**(4), 1237–1261 (2021) <https://doi.org/10.1007/s10614-020-10013-5>
- [7] Malladi, R.K.: Application of supervised machine learning techniques to forecast the COVID-19 U.S. recession and stock market crash. *Computational Economics* **63**(3), 1021–1045 (2024) <https://doi.org/10.1007/s10614-022-10333-8>
- [8] Hu, W., Zhou, J.: Trading signal survival analysis: A framework for enhancing technical analysis strategies in stock markets. *Computational Economics* **64**(6), 3473–3507 (2024) <https://doi.org/10.1007/s10614-024-10567-8>
- [9] Darwish, M., Hassanien, E.E., Eissa, A.H.B.: Stock market forecasting: From traditional predictive models to large language models. *Computational Economics* (2025) <https://doi.org/10.1007/s10614-025-11024-w>
- [10] Li, L., Chu, W., Langford, J., Schapire, R.E.: A contextual-bandit approach to personalized news article recommendation. In: *Proceedings of the 19th International Conference on World Wide Web*, pp. 661–670 (2010). <https://doi.org/10.1145/1772690.1772758>
- [11] Agrawal, S., Goyal, N.: Thompson sampling for contextual bandits with linear payoffs. In: *Proceedings of the 30th International Conference on Machine Learning*, pp. 127–135 (2013)
- [12] Zhou, D., Li, L., Gu, Q.: Neural contextual bandits with UCB-based exploration. In: *Proceedings of the 37th International Conference on Machine Learning*, pp. 11492–11502 (2020)
- [13] Zhang, W., Zhou, D., Li, L., Gu, Q.: Neural thompson sampling. In: *International Conference on Learning Representations* (2021)
- [14] Hamilton, J.D.: A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica* **57**(2), 357–384 (1989) <https://doi.org/10.2307/1912559>
- [15] Kim, C.-J., Nelson, C.R.: *State-Space Models with Regime Switching: Classical*

- and Gibbs-Sampling Approaches. MIT Press, ??? (1999)
- [16] Chen, Y., Lee, C.-W., Luo, H., Wei, C.-Y.: A new algorithm for non-stationary contextual bandits: Efficient, optimal, and parameter-free. In: Proceedings of the 32nd Conference on Learning Theory, pp. 1–30 (2019)
 - [17] Ding, Q., Hsieh, C.-J., Sharpnack, J.: Multiscale Non-stationary Stochastic Bandits (2020)
 - [18] Shen, W., *et al.*: Reinforcement learning for quantitative trading. *Expert Systems with Applications* **164**, 113820 (2021)
 - [19] Cao, J., *et al.*: Deep reinforcement learning for option pricing. *Journal of Financial Economics* (2019)
 - [20] Lu, T., *et al.*: Non-stationary stochastic bandits. *Journal of Machine Learning Research* **22**, 1–67 (2021)
 - [21] Chakraborty, S.: Market regime identification and forecasting using machine learning methods. *Applied Economics Letters* (2021)
 - [22] Fenn, D., *et al.*: Hierarchical clustering of financial time series. *Physical Review E* (2012)
 - [23] Marti, G., *et al.*: Clustering financial time series. *Wilmott* (2017)
 - [24] Kohonen, T.: *Exploration of Very Large Databases by Self-Organizing Maps*. Elsevier, ??? (1997)
 - [25] Ang, A., Timmermann, A.: Volatility timing, market timing, and short-term uncertainty. *Journal of Financial Economics* (2012)
 - [26] Fischer, T., Krauss, C.: Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research* **270**(2), 654–669 (2018) <https://doi.org/10.1016/j.ejor.2017.11.054>
 - [27] Tsantekidis, A., Passalis, N., Tefas, A., Kannianen, J., Gabbouj, M., Iosifidis, A.: Forecasting stock prices from the limit order book using convolutional neural networks. In: Proceedings of the IEEE Conference on Engineering Applications of Neural Networks (2017)
 - [28] Gandhudi, M., Alphonse, P.J.A., Fiore, U., Gangadharan, G.R.: Explainable hybrid quantum neural networks for analyzing the influence of tweets on stock price prediction. *Computers and Electrical Engineering* **118**, 109302 (2024) <https://doi.org/10.1016/j.compeleceng.2024.109302>
 - [29] Mnih, V., *et al.*: Human-level control through deep reinforcement learning. *Nature* **518**(7540), 529–533 (2015) <https://doi.org/10.1038/nature14236>

- [30] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. In: International Conference on Learning Representations (2017)
- [31] Liang, Z., et al.: Adversarial deep reinforcement learning in portfolio management. arXiv preprint arXiv:1808.09940 (2018)