Adaptive Trading Strategy Design via Greedy Search and LSTM Optimization

Dionisios N. Sotiropoulos

1. Long and Short Positions

1.1. Setup and Assumptions

We consider a stock whose price evolves over time, represented by the time series $P(t)$ for $t \in [t_{\min}, t_{\max}]$. At a given time $t_0 \in [t_{\min}, t_{\max}]$, the trader has available capital denoted by $C(t_0)$. We assume that all trades incur a fixed transaction cost $f > 0$. No leverage or margin trading is permitted, which implies that all positions must be financed using available capital. Any position, whether long or short, must be closed at some later time $t > t_0$.

1.2. Long Position: Entry Constraint Derivation

To initiate a long position at time t_0, the trader purchases q shares at price $P(t_0)$, incurring a total purchase cost of $q \cdot P(t_0)$. Including the transaction fee f, the total capital required to open the position becomes $q \cdot P(t_0) + f$. To ensure feasibility under the available capital constraint, this total must not exceed $C(t_0)$:

$$q \cdot P(t_0) + f \leq C(t_0)$$

Solving for q, we obtain the maximum number of shares:

$$q \leq \frac{C(t_0) - f}{P(t_0)} \quad \Rightarrow \quad q_{\text{long}}^{\max} = \left\lfloor \frac{C(t_0) - f}{P(t_0)} \right\rfloor$$

A valid long position size must satisfy $q \in \{0, 1, \ldots, q_{\text{long}}^{\max}\}$.

1.3. Long Position: Profit Derivation

Suppose a long position of size q (with $q \leq q_{\text{long}}^{\max}$) is opened at time t_0 and closed at time $t > t_0$. The trader initially pays $q \cdot P(t_0) + f$ to buy the shares. At the closing time t, the shares are sold at price $P(t)$, yielding a revenue of $qP(t)$ and incurring an additional transaction cost f. The net profit from this transaction is therefore:

$$\begin{aligned}
\text{Initial Outflow} &= q \cdot P(t_0) + f \\
\text{Final Inflow} &= q \cdot P(t) - f \\
\text{Profit}_{\text{long}}(t_0, t) &= q \cdot (P(t) - P(t_0)) - 2f
\end{aligned}$$

This expression captures both the price appreciation and the cost structure of the round-trip transaction.

1.4. Short Position: Entry Constraint Derivation

To initiate a short position, the trader borrows q shares and sells them at the market price $P(t_0)$, receiving proceeds of $qP(t_0)$. After subtracting the transaction fee f, the net initial inflow is $q \cdot P(t_0) - f$. However, to close the short position, the trader must repurchase q shares. To be conservative, we impose a constraint ensuring that the trader has sufficient capital to repurchase the shares at a worst-case future price $P_{\text{worst}} = \lambda \cdot P(t_0)$ for $\lambda > 1$ and also accounting for the closing fee f. Thus, the feasibility condition is given by:

$$q \cdot P_{\text{worst}} + f \leq C(t_0)$$

Solving for q, we derive the maximum allowable short position:

$$q \leq \frac{C(t_0) - f}{P_{\text{worst}}} \quad \Rightarrow \quad q_{\text{short}}^{\max} = \left\lfloor \frac{C(t_0) - f}{P_{\text{worst}}} \right\rfloor$$

1.5. Short Position: Profit Derivation

Let q be the number of shares shorted at time t_0 (with $q \leq q_{\text{short}}^{\max}$), and assume the position is closed at $t > t_0$. The initial cash inflow from selling the borrowed shares is $qP(t_0)$, reduced by the fee f. At time t, the trader buys back the q shares at price $P(t)$ and pays another fee f. The net profit is:

$$\begin{aligned}
\text{Initial Inflow} &= q \cdot P(t_0) - f \\
\text{Final Outflow} &= q \cdot P(t) + f \\
\text{Profit}_{\text{short}}(t_0, t) &= q \cdot (P(t_0) - P(t)) - 2f
\end{aligned}$$

This profit expression reflects the gain from a price decline during the holding period, net of transaction costs.

2. Technical Indicators

In financial time series analysis, technical indicators are commonly used to detect market trends, momentum shifts, volatility regimes, and trading volume anomalies. All indicators discussed below are computed using a rolling window of the most recent n time steps, ensuring that only past information is used at any point in time.

2.1. Moving Averages (MA)

Moving averages are fundamental tools in trend analysis. They smooth out short-term fluctuations and help identify the underlying direction of asset prices.

The **Simple Moving Average (SMA)** is calculated as the unweighted average of the past n prices:

$$\text{SMA}_n(t) = \frac{1}{n} \sum_{i=0}^{n-1} P(t-i).$$

In contrast, the **Exponential Moving Average (EMA)** assigns exponentially decreasing weights to older observations. It is computed recursively as:

$$\mathrm{EMA}_n(t) = \alpha P(t) + (1-\alpha)\mathrm{EMA}_n(t-1), \quad \text{with } \alpha = \frac{2}{n+1}.$$

These averages are used to generate trading signals. Specifically, when a short-term moving average $\mathrm{MA}_{\text{short}}$ crosses above a long-term moving average $\mathrm{MA}_{\text{long}}$, it is referred to as a *Golden Cross*, indicating potential upward momentum and suggesting a long entry. Formally, this occurs when:

$$\mathrm{MA}_{\text{short}}(t) > \mathrm{MA}_{\text{long}}(t), \quad \mathrm{MA}_{\text{short}}(t-1) \leq \mathrm{MA}_{\text{long}}(t-1).$$

Conversely, a *Death Cross*—used as a short entry or long exit signal—occurs when:

$$\mathrm{MA}_{\text{short}}(t) < \mathrm{MA}_{\text{long}}(t), \quad \mathrm{MA}_{\text{short}}(t-1) \geq \mathrm{MA}_{\text{long}}(t-1).$$

Reference: Brock et al. (1992), *Journal of Financial and Quantitative Analysis*.

2.2. Relative Strength Index (RSI)

The **Relative Strength Index (RSI)** is a bounded momentum oscillator that quantifies the speed and magnitude of recent price changes. It is defined as:

$$\mathrm{RSI}(t) = 100 - \left(\frac{100}{1 + RS(t)}\right),$$

where $RS(t)$ is the ratio of average gains to average losses over the past n periods:

$$RS(t) = \frac{\text{Average Gain}(t)}{\text{Average Loss}(t)}.$$

To compute these averages, define the daily changes:

$$G_i = \max(P(i) - P(i-1), 0), \quad L_i = \max(P(i-1) - P(i), 0).$$

Then:

$$\text{Average Gain}(t) = \frac{1}{n}\sum_{i=t-n+1}^{t} G_i, \quad \text{Average Loss}(t) = \frac{1}{n}\sum_{i=t-n+1}^{t} L_i.$$

Interpretation of RSI values typically follows:

- **RSI** < 30: The asset may be oversold, implying a potential upward reversal (buy signal).
- **RSI** > 70: The asset may be overbought, suggesting a potential downward correction (sell signal).

Reference: Wilder, J. W. (1978), *New Concepts in Technical Trading Systems*.

2.3. MACD and Signal Line

The **Moving Average Convergence Divergence (MACD)** is a momentum indicator that tracks the difference between two EMAs:

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

To enhance signal reliability, a **Signal Line** is computed as a 9-period EMA of the MACD:

$$\text{Signal}(t) = \text{EMA}_9(\text{MACD}(t)).$$

Trading signals are generated from the crossover of MACD and its signal line:

- **Bullish Signal (Buy):** $\text{MACD}(t) > \text{Signal}(t)$ and $\text{MACD}(t-1) \leq \text{Signal}(t-1)$.
- **Bearish Signal (Sell):** $\text{MACD}(t) < \text{Signal}(t)$ and $\text{MACD}(t-1) \geq \text{Signal}(t-1)$.

The MACD effectively captures shifts in trend momentum and potential reversals when divergence from price trends is observed.

Reference: Gerald Appel, *Technical Analysis: Power Tools for Active Investors*.

2.4. Bollinger Bands

Bollinger Bands provide a dynamic range for price movement based on recent volatility. They consist of:

- A middle band: an SMA of the asset price,
- An upper band: the SMA plus k times the standard deviation $\sigma_n(t)$,
- A lower band: the SMA minus k times $\sigma_n(t)$.

Formally:

$$\begin{aligned}
\text{SMA}_n(t) &= \frac{1}{n}\sum_{i=0}^{n-1} P(t-i), \\
\sigma_n(t) &= \sqrt{\frac{1}{n}\sum_{i=0}^{n-1}(P(t-i) - \text{SMA}_n(t))^2}, \\
\text{Upper}(t) &= \text{SMA}_n(t) + k \cdot \sigma_n(t), \\
\text{Lower}(t) &= \text{SMA}_n(t) - k \cdot \sigma_n(t).
\end{aligned}$$

Interpretation:

- Wide bands reflect higher volatility; narrow bands suggest stability.
- Price near or above the upper band indicates potential overbought conditions.
- Price near or below the lower band suggests possible oversold conditions.

Common trading signals include price reversion when breaching a band or confirmation of trend continuation based on band interactions.

Reference: Bollinger, J. (2001), *Bollinger on Bollinger Bands*.

2.5. On-Balance Volume (OBV)

The **On-Balance Volume (OBV)** indicator is a cumulative measure that uses trading volume to infer buying and selling pressure. It updates as:

$$\mathrm{OBV}(t) = \mathrm{OBV}(t-1) + \mathrm{sgn}(P(t) - P(t-1)) \cdot V(t),$$

where $V(t)$ is the trading volume at time t and:

$$\mathrm{sgn}(x) = \begin{cases} +1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases}.$$

Interpretation:

- OBV rising with price confirms an uptrend.
- OBV falling with price supports a downtrend.
- Divergence between OBV and price may indicate weakening trends or impending reversals.

OBV is best used alongside price action and trend-following strategies for confirmation.
Reference: Granville, J. (1963), *Granville's New Key to Stock Market Profits*.

3. Fundamental Indicators

3.1. Price-to-Earnings Ratio

Let $\mathcal{T} = \{t_{\min}, \ldots, t_{\max}\}$ denote the total trading horizon, and let $\mathcal{T}_{\mathrm{EPS}} = \{t_r\} \subseteq \mathcal{T}$ denote the set of discrete time points at which the trading algorithm generates estimated earnings per share (EPS) values, denoted by $E(t_r)$. These earnings may be obtained from historical market data or predicted via time series models (e.g., LSTM), and are used internally by the strategy as proxies for firm fundamentals.

For any time $t \in \mathcal{T}$, define $t^{\mathrm{last}}(t) = \max\{t_r \in \mathcal{T}_{\mathrm{EPS}} \mid t_r \leq t\}$ as the most recent earnings report date available at time t. The trailing Price-to-Earnings ratio is defined as:

$$\mathrm{P/E}(t) = \frac{P(t)}{E(t^{\mathrm{last}}(t))},$$

where $P(t)$ denotes the stock's closing price at time t.

Interpretation. The P/E ratio measures how much investors are willing to pay per unit of trailing earnings:

- A **low P/E** may signal undervaluation or weak expected growth.
- A **high P/E** may indicate overvaluation or strong growth expectations.

Integration into Trading Algorithms.

- **In Question 1:** The P/E ratio is treated as a precomputed indicator over rolling windows $[t - L_i + 1, t]$ using the most recent earnings report at each time.
- **In Question 2:** A model-estimated EPS $\hat{E}(t)$ is computed daily using a forecasting function f_{EPS}, allowing P/E to be defined continuously as:

$$\text{P/E}(t) = \frac{P(t)}{\hat{E}(t)}.$$

  These values are used for precomputation over candidate windows $\mathcal{L}_i$ and integrated into the LSTM-based optimization pipeline.

Reference. Basu, S. (1977). *Investment Performance of Common Stocks in Relation to Their Price–Earnings Ratios.* Journal of Finance.

3.2. Earnings Surprise

Definition. Earnings surprise quantifies the deviation between actual and expected earnings:

$$\text{Surprise}(t) = \frac{E_{\text{actual}}(t) - E_{\text{expected}}(t)}{E_{\text{expected}}(t)},$$

where $E_{\text{expected}}(t)$ is predicted by a time series model trained on historical earnings.

Model-Based Forecasting.

$$E_{\text{expected}}(t) = f\left(E(t-1), E(t-2), \ldots, E(t-k); \theta\right),$$

where:

- f is a forecasting model such as an LSTM, regression tree, or temporal convolutional network,
- k is the look-back window,
- θ are the learnable parameters of the model.

Interpretation.

- A **positive surprise** ($\text{Surprise}(t) > 0$) may imply outperformance and future price increase.
- A **negative surprise** ($\text{Surprise}(t) < 0$) may imply underperformance and possible price decline.

Usage in Algorithms.

- **In Question 1 (Greedy Optimization):** Earnings surprise values are only available at discrete reporting dates $t \in \mathcal{T}_{\text{EPS}}$. They are incorporated by evaluating whether Surprise(t) crosses a user-defined threshold. If the value is above a positive threshold, it may trigger a long entry signal; if it falls below a negative threshold, it may suggest a short entry. Since the values are sparse, they are only used when actual earnings data is available.
- **In Question 2 (LSTM-Based Adaptive Model):** The surprise signal is incorporated in one of the following ways:
  - *Forward-filling:* The most recent value of Surprise(t) is carried forward to subsequent days until a new report arrives, allowing its use at every time t.
  - *Smoothing:* The signal is smoothed using exponential decay or moving averages to reduce discontinuities.

  These approaches allow the model to integrate the earnings surprise information even though it is not available daily, making it compatible with a continuous-time trading framework.

Reference. Bernard and Thomas (1989), *Post-Earnings-Announcement Drift.*

4. Question 1: Greedy Optimization of Technical and Fundamental Indicator Parameters

This question focuses on developing an agent that performs greedy optimization of both technical and fundamental indicator parameters to maximize cumulative returns during a training period, using only historically available data. The optimization involves determining the best combination of time window lengths and signal threshold values for both types of indicators under realistic trading constraints.

4.1. Objective

The goal is to construct a trading agent that:

- Learns optimal past time window sizes L for computing each indicator,
- Determines optimal thresholds θ_i that define trading signals for both technical and fundamental indicators,
- Makes trading decisions (long, short, close) to maximize total profit,
- Evaluates performance using the Sharpe ratio.

4.2. Setup and Constraints

- Let the total time horizon be $\mathcal{T} = \{t_{\min}, \ldots, t_{\max}\}$.
- Partition $\mathcal{T}$ into a training set $\mathcal{T}_{\text{train}} = \{t_1, \ldots, t_K\}$ and a testing set $\mathcal{T}_{\text{test}} = \{t_{K+1}, \ldots, t_{\max}\}$.
- The agent begins with an initial capital $C(t_0)$ at $t_0 = t_1$.
- No leverage or margin trading is allowed.
- Each transaction incurs a fixed fee $f > 0$.
- Trading positions (long or short) must be feasible and follow the constraints on capital availability as described in Sections 1.2 to 1.5.
- Indicators available to the agent include:
  - **Technical:** SMA, EMA, RSI, MACD, Bollinger Bands, OBV
  - **Fundamental:** P/E Ratio, Earnings Surprise

4.3. Indicator Computation and Signal Generation

Let $I_i(t; L_i)$ denote the value of the i-th indicator computed over the past window $[t-L_i+1, t]$. Each indicator has a parameter vector $\theta_i = (\theta_i^+, \theta_i^-)$ encoding the thresholds for upward and downward signals.

The signal function $\mathcal{S}_i(t; L_i, \theta_i) \in \{-1, 0, +1\}$ is defined as:

$$\mathcal{S}_i(t; L_i, \theta_i) = \begin{cases} +1 & \text{if } I_i(t; L_i) > \theta_i^+ \\ -1 & \text{if } I_i(t; L_i) < \theta_i^- \\ 0 & \text{otherwise} \end{cases}$$

The global trading decision $\mathcal{D}(t)$ is determined by aggregating the individual signals:

$$\mathcal{D}(t) = \text{Aggregate}(\{\mathcal{S}_i(t; L_i, \theta_i)\}_{i=1}^N),$$

where Aggregate$(\cdot)$ is a rule such as majority vote or weighted sum.

Position functions:

- **Entry:** A position is entered at time t if $\mathcal{D}(t) \in \{-1, +1\}$ and no open position currently exists.
- **Exit:** An open position is closed at time t if $\mathcal{D}(t) = 0$ or $\mathcal{D}(t)$ signals the opposite direction.

4.4. Trading Algorithm

At each time t in $\mathcal{T}_{\text{train}}$, the agent performs the following steps:

1. Compute indicators $I_i(t; L_i)$ for all i.
2. Compute $\mathcal{S}_i(t; L_i, \theta_i)$ and aggregate into a global decision $\mathcal{D}(t)$.
3. Execute one of:
   - Open long/short position (if none is open and feasible).
   - Close current position if indicated.
4. Check capital constraints, update holdings and capital $C(t+1)$.

4.5. Performance Evaluation and Sharpe Ratio

Define return at each time t as:

$$R(t) = \frac{C(t) - C(t-1)}{C(t-1)}.$$

The Sharpe ratio on a period $\mathcal{T}$ ($\mathcal{T}_{\text{eval}}$ or $\mathcal{T}_{\text{train}}$) is computed as:

$$\text{Sharpe Ratio} = \frac{\mu_R}{\sigma_R},$$

where:

$$\mu_R = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} R(t),$$

$$\sigma_R = \sqrt{\frac{1}{|\mathcal{T}| - 1} \sum_{t \in \mathcal{T}} (R(t) - \mu_R)^2}.$$

Assume the risk-free rate $R_f = 0$ for simplicity.

4.6. Optimization Procedure

1. Define candidate values for:
   - Window sizes $L_i \in \{L_1, ..., L_m\}$
   - Threshold pairs $\theta_i = (\theta_i^+, \theta_i^-)$ for each indicator.
2. For each parameter configuration:
   (a) Run the trading strategy on $\mathcal{T}_{\text{train}}$.
   (b) Compute final capital $C(t_K)$ and Sharpe ratio.
3. Choose the parameter set maximizing Sharpe ratio on $\mathcal{T}_{\text{train}}$.
4. Test this configuration on $\mathcal{T}_{\text{test}}$ to assess generalization.

4.7. Model Architecture and Signal Generation

The overall architecture of the LSTM-based Trading Agent appear in Figure 1. At each time $t \in \mathcal{T}$, the trading agent performs the following steps:

1. **LSTM Feature Encoding.** Let $\lambda_{\max} \in \mathbb{N}$ be the fixed input sequence length. The agent constructs a past sequence of raw market features:

$$X_t = \{x_{t-\lambda_{\max}+1}, \ldots, x_t\} \in \mathbb{R}^{\lambda_{\max} \times d_{\text{in}}},$$

   where each $x_\tau \in \mathbb{R}^{d_{\text{in}}}$ contains raw features such as prices, returns, or volumes. This sequence does *not* contain precomputed indicators.

   A shared LSTM encoder processes X_t and outputs a context vector $h_t \in \mathbb{R}^d$ (the final hidden state), which serves as a compressed summary of the recent market history.

2. **Per-Indicator Parameterization.** For each indicator $i \in \{1, \ldots, N\}$, the model applies indicator-specific linear layers to compute the following:

   - **Trainable parameters:**
     - $W_i^w \in \mathbb{R}^{M \times d}$ and $b_i^w \in \mathbb{R}^M$ — parameters used to produce window size logits.
     - $W_i^\theta \in \mathbb{R}^{2 \times d}$ and $b_i^\theta \in \mathbb{R}^2$ — parameters used to predict signal thresholds.
   - **Window Size Logits:**

$$\alpha_i(t) = W_i^w h_t + b_i^w \in \mathbb{R}^M,$$

     where M is the number of predefined candidate window sizes $\Lambda_i = \{\lambda_i^{(1)}, \ldots, \lambda_i^{(M)}\}$.
   - **Gumbel-Softmax Sampling:** To approximate discrete window selection, a soft categorical distribution is constructed:

$$w_i^{(m)}(t) = \frac{\exp((\log \alpha_i^{(m)} + g_m)/\tau)}{\sum_{j=1}^M \exp((\log \alpha_i^{(j)} + g_j)/\tau)}, \quad g_j \sim \text{Gumbel}(0, 1),$$

     yielding a soft attention vector $w_i(t) \in \mathbb{R}^M$ lying in the probability simplex Δ^{M-1}.
   - **Soft Indicator Computation:** Precompute $I_i(t; \lambda_i^{(m)})$ for each window size $m = 1, \ldots, M$. The LSTM-generated weights determine the indicator used at time t:

$$\hat{I}_i(t) = \sum_{m=1}^{M} w_i^{(m)}(t) \cdot I_i(t; \lambda_i^{(m)}).$$

   - **Threshold Prediction:** A linear projection predicts both the upper and lower signal thresholds:

$$(\theta_i^+(t), \theta_i^-(t)) = W_i^\theta h_t + b_i^\theta \in \mathbb{R}^2.$$

   - **Signal Generation:** The signal for indicator i is computed using smoothed thresholds:

$$\mathcal{S}_i(t) = \sigma_\epsilon(\hat{I}_i(t) - \theta_i^+(t)) - \sigma_\epsilon(\theta_i^-(t) - \hat{I}_i(t)),$$

     where $\sigma_\epsilon(x) = \frac{1}{1+e^{-x/\epsilon}}$ is a smooth sigmoid approximation of the step function.

3. **Global Signal Aggregation.** The per-indicator signals $\mathcal{S}_1(t), \ldots, \mathcal{S}_N(t)$ are aggregated into a final decision signal:

$$\mathcal{D}(t) = \tanh\left(\sum_{i=1}^{N} \beta_i \mathcal{S}_i(t) + b\right),$$

where $\beta_i \in \mathbb{R}$ and $b \in \mathbb{R}$ are additional trainable aggregation parameters.

4. **Trade Execution and Capital Update.** The final decision $\mathcal{D}(t)$ is interpreted as:

   - $\mathcal{D}(t) > \delta$: initiate or maintain a long position,
   - $\mathcal{D}(t) < -\delta$: initiate or maintain a short position,
   - $|\mathcal{D}(t)| \leq \delta$: hold position or close any existing one.

   The portfolio is updated accordingly, and capital evolves as $C(t+1)$ under the constraints of Sections 1.2 to 1.5.

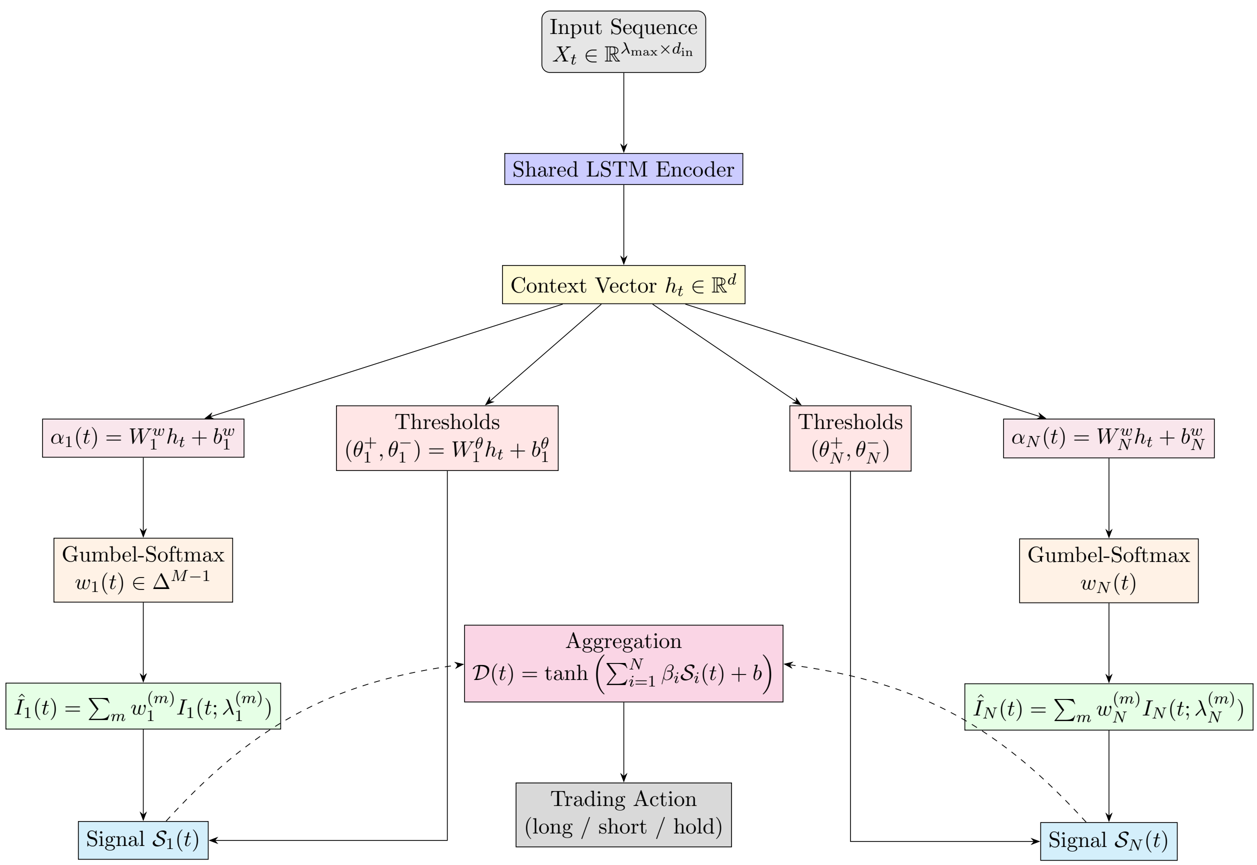

Figure 1: LSTM-Based Trading Agent with Indicator-Specific Heads and Aggregation (scaled to page width).

4.8. Training Objective

The model is trained on $\mathcal{T}_{\text{train}}$ using differentiable portfolio-based objectives:

- Sharpe Ratio Loss:

$$R(t) = \frac{C(t) - C(t-1)}{C(t-1)}, \quad \mathcal{L}_{\text{Sharpe}} = -\frac{\mu_R}{\sigma_R}$$

- Cumulative Return Loss: $\mathcal{L}_{\text{ret}} = -\sum_t R(t)$
- Hybrid Loss: $\mathcal{L} = \mu_{\text{ret}}\mathcal{L}_{\text{ret}} + \mu_{\text{Sharpe}}\mathcal{L}_{\text{Sharpe}} + \mu_{\text{reg}}\|\theta\|^2$

4.9. Evaluation and Comparison

Evaluate the model on $\mathcal{T}_{\text{test}}$:

- Report Sharpe ratio, cumulative return, volatility, and drawdown
- Compare against the greedy optimization baseline from Question 1

5. Deliverables and Submission Guidelines

Group Work Policy

Students may work individually or in groups of up to **three (3)** members. All members of the group are expected to contribute meaningfully to both implementation and analysis. Only one submission per group is required, and the names and student IDs of all group members must be clearly indicated.

Deliverables for Question 1 (Greedy Optimization)

- **Code implementation** of the greedy optimization algorithm that searches over window sizes λ_i and thresholds θ_i for all indicators.
- **Tables of results** summarizing optimal parameter configurations found during training and corresponding test set performance (Sharpe ratio, cumulative return, volatility, drawdown).
- **Plots** of:
  - Portfolio value over time for both training and test periods,
  - Entry/exit points on price time series (optional but recommended).
- **Extended report** explaining:
  - The choice of indicator sets and trading logic,
  - The grid search strategy used for optimization,
  - Interpretation of experimental results and model behavior.

Deliverables for Question 2 (LSTM-Based Adaptive Optimization)

- **Code implementation** of the end-to-end LSTM-based architecture, including:
  - Raw input processing (past prices/volumes),
  - Indicator parameter prediction (window sizes $\lambda_i(t)$ and thresholds $\theta_i(t)$),
  - Trading signal generation and portfolio update mechanism,
  - Training loop with differentiable Sharpe or return-based loss.
- **Tables of results** comparing the LSTM-based model with the greedy baseline from Question 1, across the same performance metrics.
- **Visualizations** of:
  - Learned dynamic window sizes $\lambda_i(t)$ and thresholds $\theta_i(t)$ over time,
  - Portfolio performance on training and test periods,
  - Optional: signal activations and decision scores $\mathcal{D}(t)$.
- **Extended technical report** describing:
  - The model architecture and training setup,
  - Hyperparameter choices (e.g., temperature τ, smoothing ϵ),
  - The rationale behind dynamic parameterization and comparison with the greedy baseline,
  - A critical analysis of the model's performance and potential improvements.

Submission Format

All code must be submitted in a single compressed folder (e.g., `.zip` or `.tar.gz`) containing:

- Well-commented and executable Python code,
- README file with instructions to reproduce results,
- PDF report(s).

All submissions must be uploaded to the course platform by the announced deadline.