**Problem Definition**

[4] Stock price data is characterized by high noise and sequential patterns. Our initial approach involved

Therefore, extracting useful data from such noise and effectively handling time-series data is crucial.  
Our previously proposed method to address this involved using **VMD** to decompose noisy stock price data into simpler signals and employing a **TCN** network to process the time-series data.

[5] However, since this model did not converge easily, we introduced a new methodology.

[6] inspiration from the methodology proposed in the referenced paper.

[7] Key changes include replacing the TCN model with an LSTM model and substituting the feature extraction process we previously aimed to achieve through VMD with Cascaded-LSTM. This modification resulted in successful model convergence.

[8] Additionally, while our originally proposed algorithms included DQN and PPO, we found that DQN was unsuitable for multi-stock trading due to the high dimensionality of the action space. Instead, we adopted actor-critic algorithms such as A2C and PPO, which are compatible with continuous action spaces.

Algorithms Used

a. A2C

Actor Critic based algorithm. stabilizes policy training by calculating the "Advantage" based on the value of the state in the Actor-Critic

b. PPO

It is also Actor Critic based algorithm. Improves training stability and efficiency by introducing clipping to prevent excessive policy updates.

**Data**

[9] Due to the high noise in stock price data, it is challenging to process this data alone. So we incorporated technical indicators such as MACD, RSI, and CCI into the state representation.

The paper used data from 30 stocks, including MACD, RSI, CCI, and ADX for each stock,number of currently held stocks, current balance -> total state space dimension 181.

( **MACD (Moving Average Convergence Divergence)**:  
MACD analyzes the difference between short-term and long-term moving averages to identify the trend direction and momentum. It is commonly used to detect trend reversals.

 **RSI (Relative Strength Index)**:  
RSI measures the ratio of gains to losses over a specific period, helping to identify overbought or oversold market conditions.

 **CCI (Commodity Channel Index)**:  
CCI evaluates how far the current price is from its historical average, aiding in the detection of overbought or oversold levels.

 **ADX (Average Directional Index)**:  
ADX measures the strength of a trend, with higher values indicating a stronger trend, regardless of its direction.)

As the number of stocks increases, the state dimension grows, necessitating a larger model. Considering the limitations of our experimental environment, we used the data from three indices (S&P500, Dow Jones, KOSPI), their respective MACD, RSI, and CCI, the number of held stocks, and the current balance to construct a state with a dimension of 16.

[10] Training data spanned from 2012.12.01 to 2021.01.01, validation data covered 2021.01.01 to 2022.01.01, and testing data ranged from 2022.01.01 to 2023.01.01.

**Environment**

[11] Model Design and Training Features (p.12)

**Model Atchitecture & Training**

[12] The model consists of three main components:

Pre-LSTM, Policy Network, Value Network

All three networks utilize LSTM.

[13] Following the basic actor-critic learning algorithm, the Pre-LSTM, like the Policy Network, was trained based on reward-driven policy gradients.

**Hyperparameter**

[14] The hyperparameters for the experiment were set as follows:

Gamma: Discount factor

Learning rate: Learning rate for the optimizer

Lamda: balancing bias and variance

Clip epsilon: threshold for policy updates

PPO epochs: Number of epochs

Minibatch size: Number of samples in each mini-batch

k: for limiting maximum buyable, sellable shares

N\_channels: input feature channels for the model

Tau:Soft update rate for target, used to align the target with main

Lamda: balancing bias and variance

Clip epsilon: threshold for policy updates

PPO epochs: Number of epochsk: for limiting

**Model & Training**

[15] To explain how the entire algorithm works, as shown in the diagram on the slide, it first concatenates the states over T days and make state sequence data for T days

And the Pre-LSTM extracts features from the state sequence data and feeds them as inputs to the Policy and Value networks, which generate output results.

The paper explains that by extracting features using Pre-LSTM, the model approximates a POMDP (Partially Observable MDP) to an MDP, which is expected to improve performance.

**Results**

[16] The A2C algorithm showed instability during training compared to PPO. On the test data, A2C showed a tendency to be highly influenced by initial conditions, such as weight initialization, achieving returns ranging from as low as 1% to as high as 10% over a one-year testing period. In contrast, PPO demonstrated relatively stable convergence and achieved an average return of approximately 10%.

Model performance can be further enhanced by including more stocks, incorporating a wider variety of technical indicators, and performing more precise hyperparameter tuning.

[17] Project Schedule