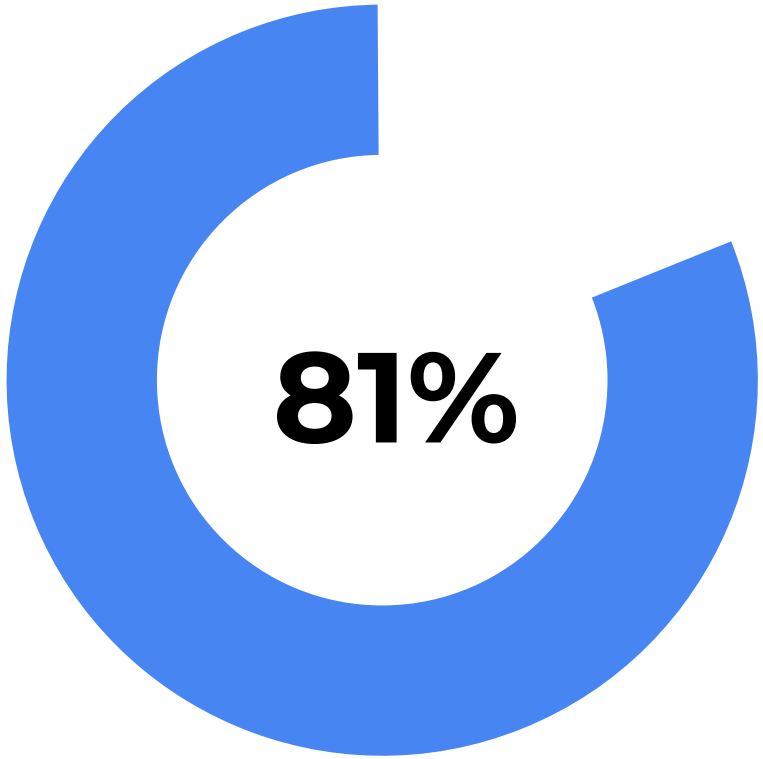
Stock Return Prediction with CNN and Attention-based LSTM Models

LIU Xiaolin
ZHANG Haoxiang
ZHANG Puming



Underperform

- 19% made positive abnormal returns
- Annually underperform 10.3%
- What about computer?

Overview

01 Data Preprocessing

02 Deep Learning Models

03 Model Evaluation

04 Conclusion

01

Preprocessing

Data Collection
Normalization
Pattern Recognition

Data Collection

- Download information from Bloomberg
- Focus on TESLA
- 6^{th} or 10^{th} day return as label
- $r_i = \frac{P_{(close,i)} - P_{(close,i-1)}}{P_{(close,i-1)}}$

Bloomberg | TESLA

Data Collection

10

Variables

5 Fundamental
5 Technical Indicators

Bloomberg

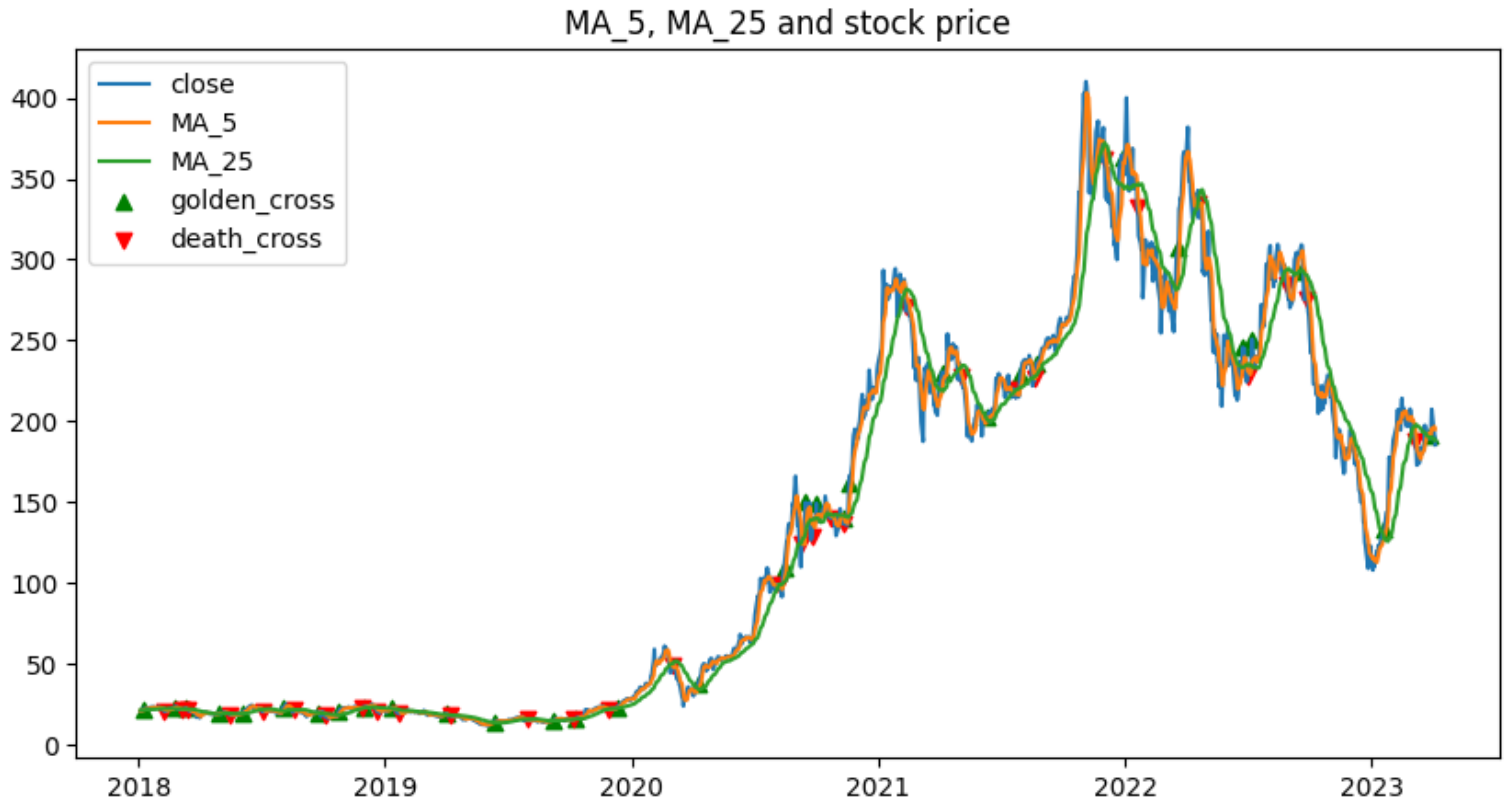
5+

Years

Starting from **2017**
Most recent data

TESLA

Pattern Recognition



Technical Patterns

- Intersection of MA(5) and MA(25)
- Golden cross & death cross

Data Collection

10

Variables

5 Fundamental
5 Technical Indicators

Bloomberg

5+

Years

Starting from **2017**
Most recent data

TESLA

Data Collection

12

Features

5 Fundamental
5 Technical Indicators
2 Technical Patterns

Bloomberg

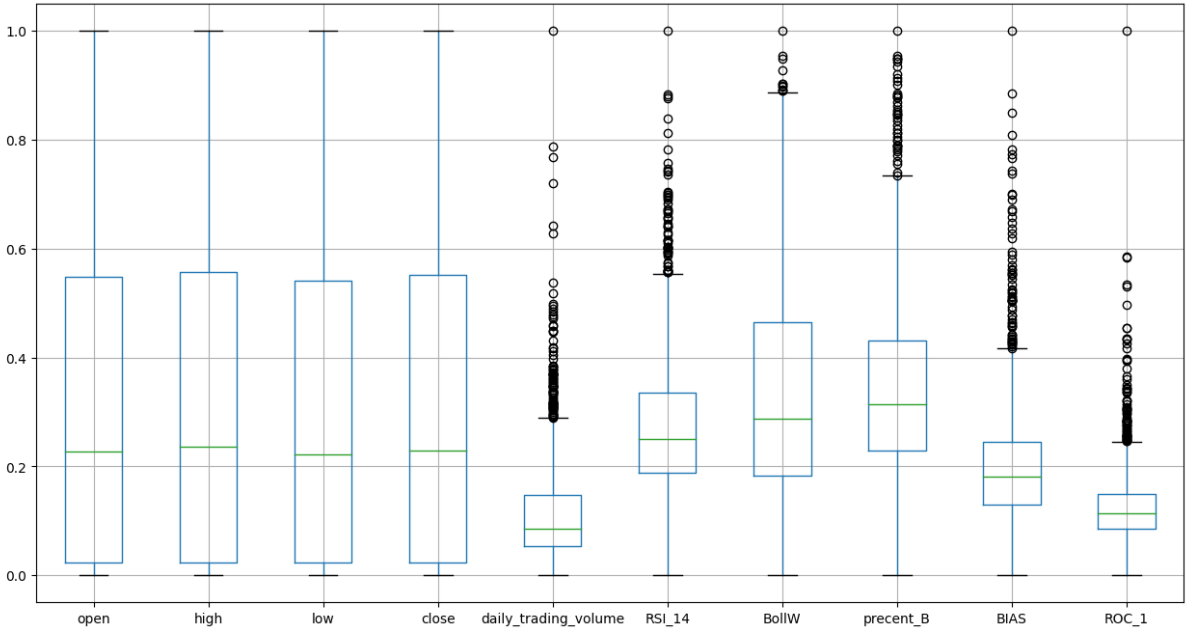
5+

Years

Starting from **2017**
Most recent data

TESLA

Normalization



Min-Max Normalization

- The largest range: 150,048,598,784
- The smallest : 0.285
- $\hat{x} = \frac{x - \min(x)}{\max(x) - \min(x)}$
- Bound between $[0,1]$

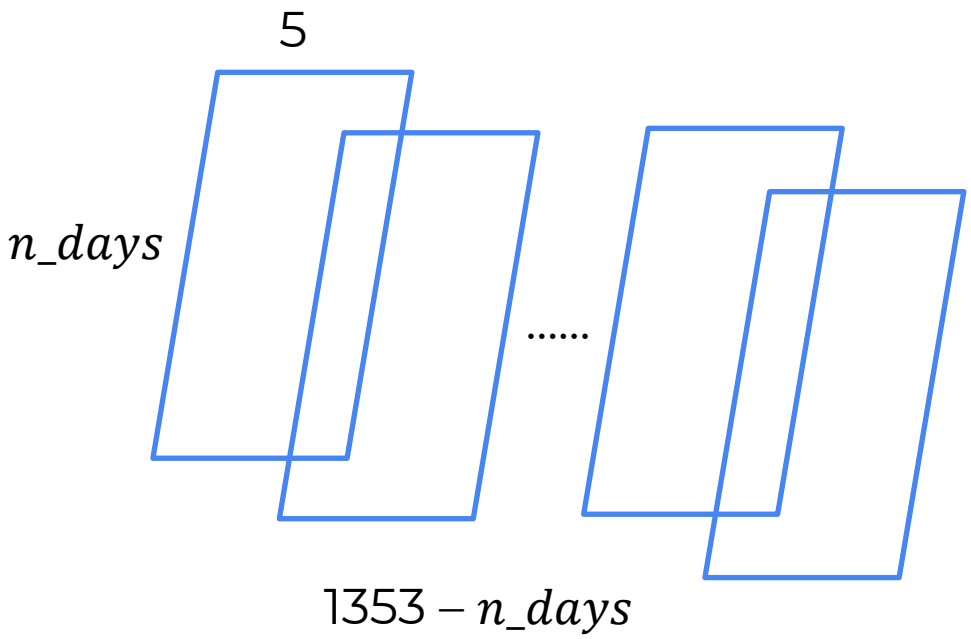
02

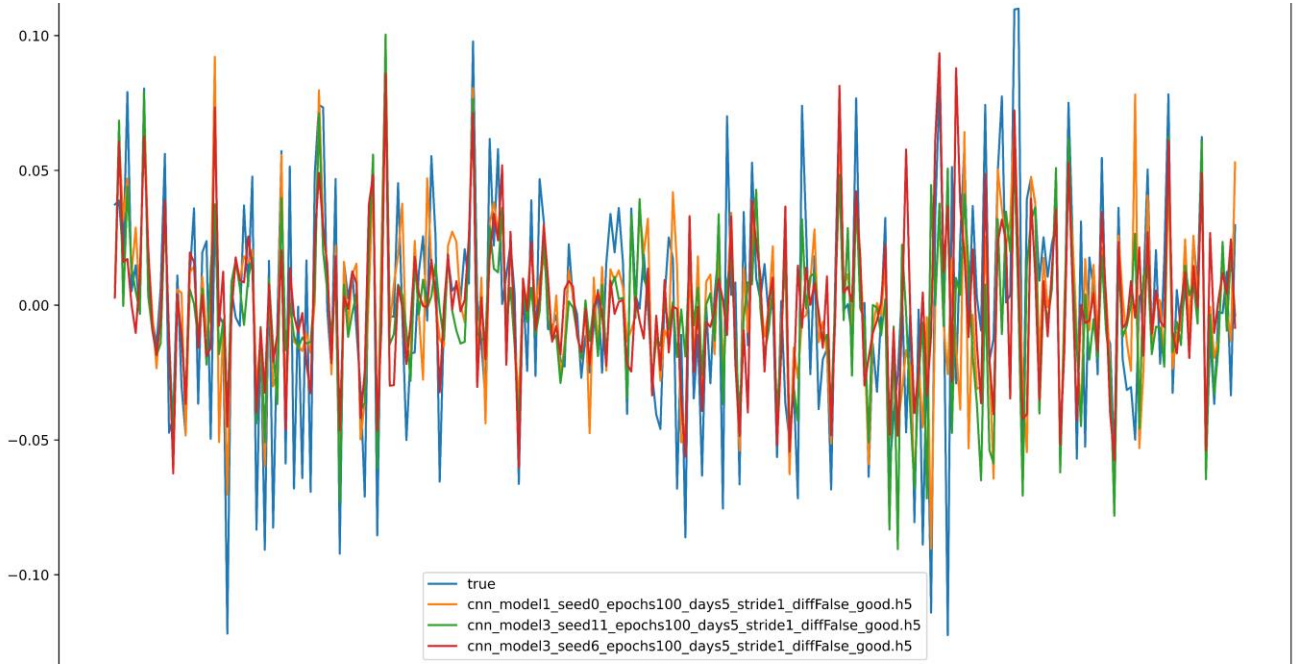
Deep Learning Models

Convolutional Neural Network (CNN)
Attention-based Long Short-Term Memory (LSTM)

CNN + MLP

- 5 days or 10 days as one observation
- $(1353 - n\_days, n\_days, 5)$ input tensor flow
- Convo | Flatten | Dense
- MSE: 0.0004



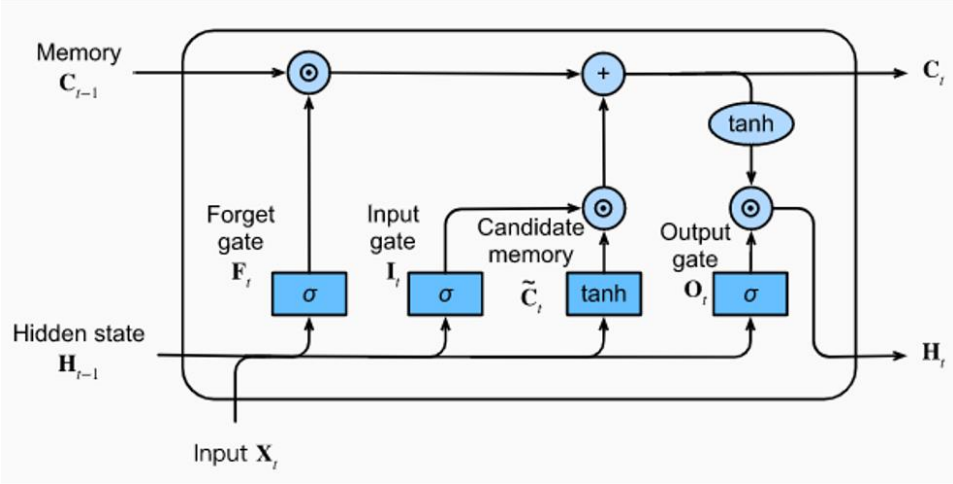


Attention-Base LSTM

Long-Short Term Memory Model

- Better at handling long-term dependencies due to their ability to remember information for extended periods of time

- less susceptible to the vanishing gradient problem with the use of LSTM cell, which helps to preserve information over long sequences

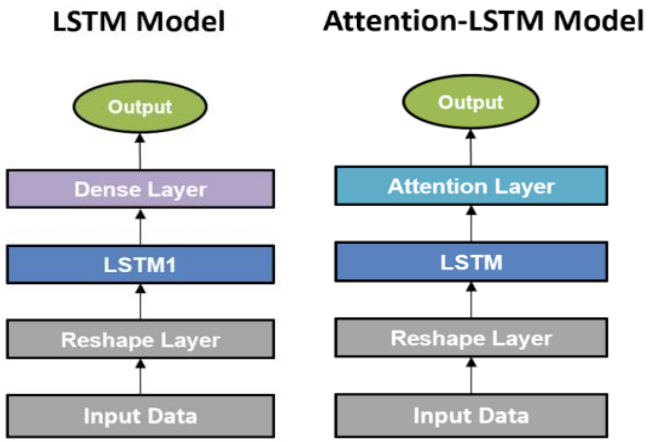


Improvement: Attention Mechanism

- Inspired by biological phenomena and human perception: Instead of treating all information with equal importance, we focuses more on those important

Extract key features and ignore the redundant features

Model Architecture



Soft Attention

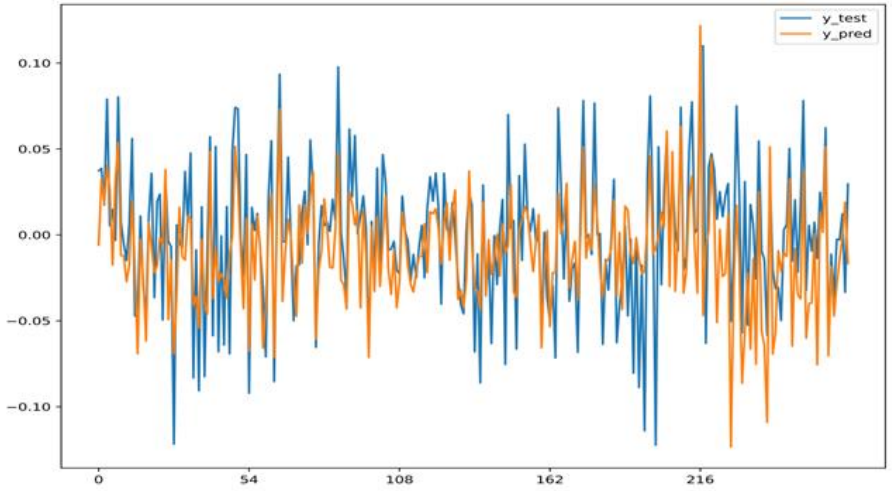
$$e_t = tanh(W_a [x_1, x_2, ..., x_T] + b) \quad \alpha_t = \frac{exp(e_t)}{\sum_{k=1}^{T} exp(e_k)}$$

W_α: Weight matrix reflecting the importance of each feature (trainable)

α_t: Resulting weights

- **Activation function**: ReLU
- **Optimization**: Adam; **Loss function**: mean squared error
- **Model Architecture**: Attention – LSTM(32) – Dense(64) – Dense(32) – Dense(1)

Model Performance



MSE

Pure LSTM	0.0025
Attention LSTM	0.00049

Back Testing

Annualized return: 106%

Attention Mechanism improved the performance

Time Series Model

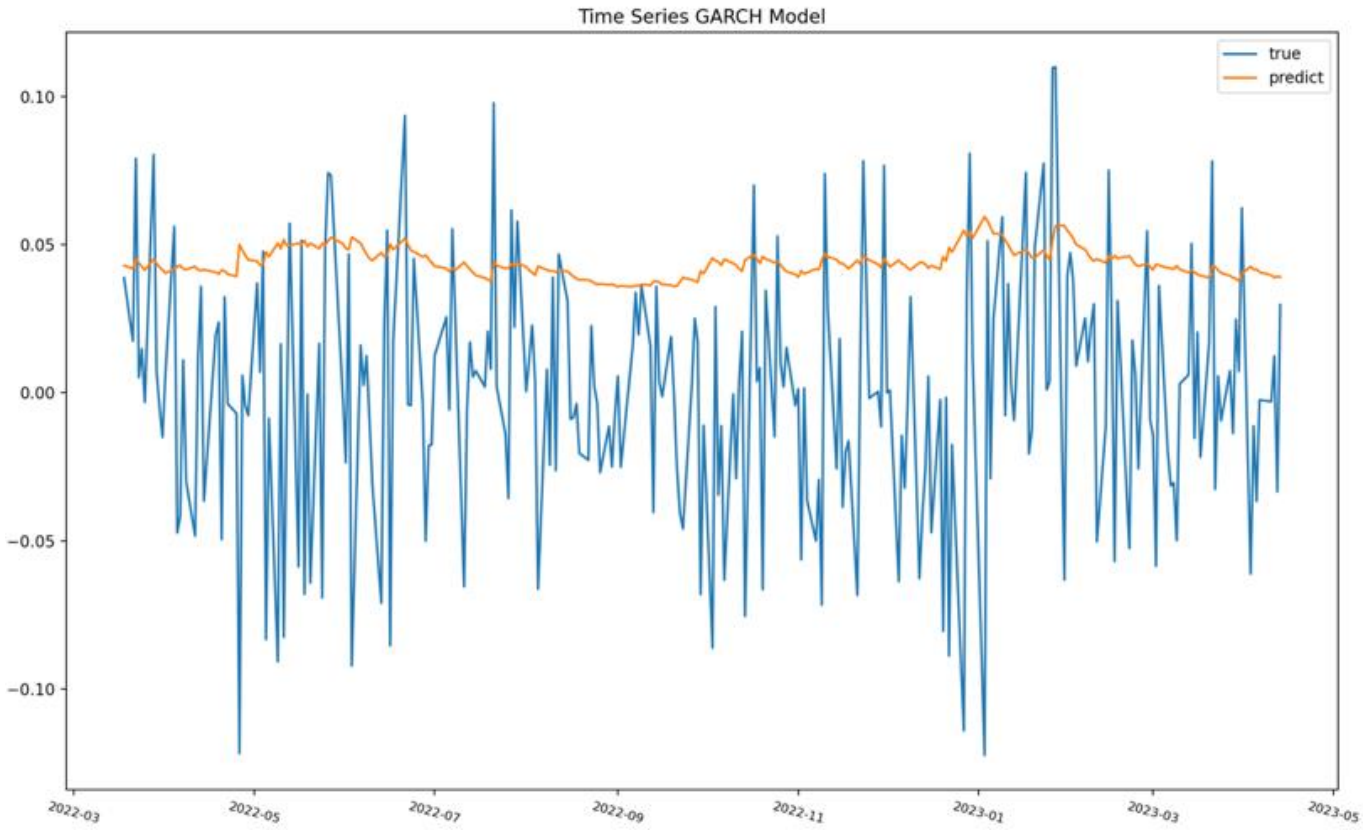
Build GARCH(1, 1) Model for Return

$$r_t = \mu + \epsilon_t$$

$$\epsilon_t = \sigma_t e_t$$

$${\sigma_t}^2 = \omega + \alpha {\epsilon_{t-1}}^2 + \beta {\sigma_{t-1}}^2$$

$$e_t \sim N(0, 1)$$



Problem

The GARCH model hardly changes
Cannot follow the variation of stock return

03
Model Evaluation

Trading Strategy

Intraday Long Only

Use previous 5 days' information to predict 6th day's return
If predicted return > 3%, go long position at Open price and close position at close price

Back Testing

Initial Capital: $100,000
Trade exposure: 60% cash
Commission fee: 0.01%

Backtrader

Evaluation Criteria

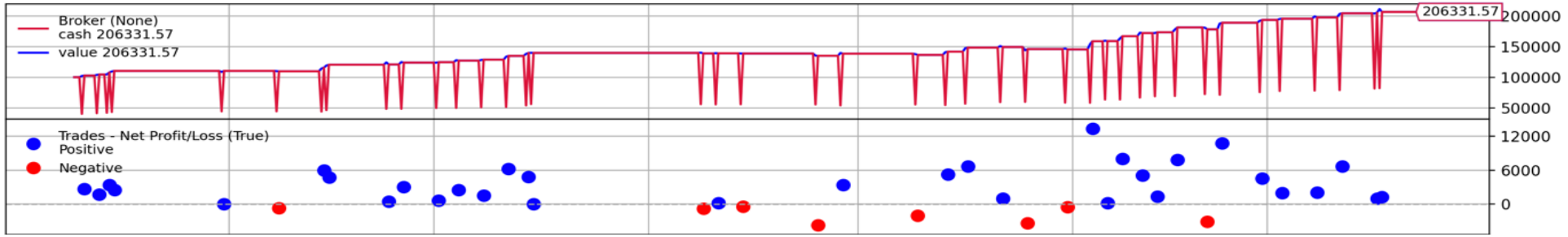
Portfolio value
Win rate

04

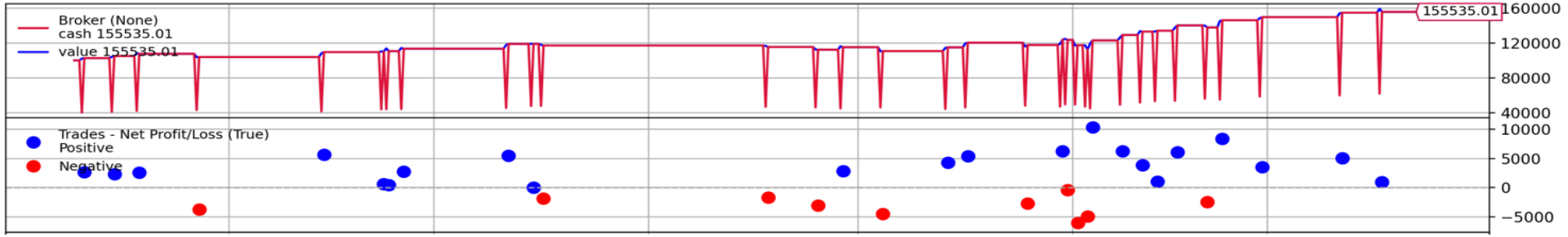
Conclusion

Conclusion

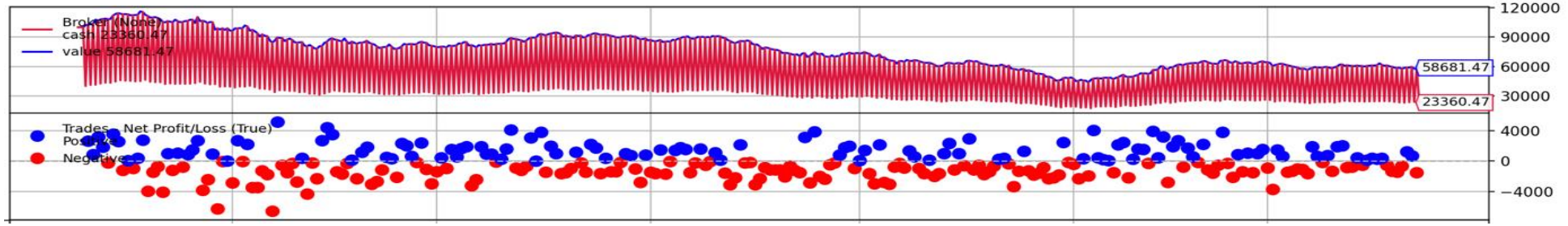
Attention-LSTM



CNN



GARCH



Q & A

Thank you!