Attention LSTM for Stock Forecasting Using Macro and Commodity Features

Xueyan Shi [1]

Abstract

Machine learning techniques have long been used in the finance sector and areas like algorithmic trading are always looking for more efficient and accurate models for better revenue. This paper presents an LSTM model with a self-attention structure that outperforms the classic LSTM anda few variants of it. Features with multiple dimensions from different sectors related to the target stock also prove to perform better compared with only using OHLCV data.

1. Introduction

Stock price prediction is a classic non-linear time-series example, which has been under the spotlight for over two centuries. Compared with traditional trading strategies, algorithmic trading with quantitative models performs better due to its subjectivity to human emotions and judgment. Traditional stock prediction models usually include Linear Regression, Autoregressive Integrated Moving Average (ARIMA), Random Forest, etc. These models have the advantage of being interpretable and lightweight, but they lack the ability to capture non-linear and time-dependent characteristics.

In the recent decade, Deep Learning with neural networks has gradually taken over in every field, including time series prediction. The individual neurons from the network are capable of learning different characteristics, making the model more robust for complex patterns and trends. Recurrent Neural Network (RNN) is a type of deep learning network that is designed for sequential data, in which the output not only depends on the current input but also the results from previous computations.

Long Short-Term Memory Networks (LSTM) is one of the most famous RNN models, with its unique gate mechanism. The forget gates are trained to determine what information should not be passed to the next neuron, enabling the algorithm to remember long-term patterns.

LSTM on stock prediction is an ongoing topic in both the research field and the finance sector. Research on different variations of LSTM shows that by augmenting LSTM, either by changing its internal structure, or combining it with other neural networks can yield better results in specific targets. For stock prediction, a common variation includes bidirectional LSTM (Schuster & Paliwal, 1997) and peephole LSTM (Gers et al., 2000), in which one tries to train data from both directions for feature reinforcement, and the other focuses on modifying the forget gate inside the LSTM neuron. In 2017, the famous attention mechanism was introduced (Vaswani et al., 2017), which allows models to dynamically assign weights to different parts of the input sequence, making it the new foundational block of time-series forecasting.

This paper mainly focuses on how to integrate the two foundational building blocks together, creating an attention-LSTM model for individual stock price prediction. The input data is first fed into a stacked LSTM and then processed by a self-attention layer. This structure is shown to perform better on the Nvidia Stock (ticker: NVDA) compared with the classic LSTM and its variant. The paper also stressed the importance of using data from macro economy and related commodities can also improve performance by adding more dimensions to the data rather than OHLCV data alone.

2. Method

2.1. Data Collection and Feature Engineering

Feature engineering was performed on the main dataset, with the following indicators calculated based on the close price of Nvidia:

- **Exponential Moving Averages (EMA)** Computed over multiple window sizes to capture trends over different time horizons.
- **Rate of Change (ROC)** Measures momentum by evaluating percentage change over window periods.
- **Relative Strength Index (RSI)** Quantifies the magni-

[1]Department of Cognitive Science, University of California San Diego, United States. Correspondence to: Xueyan Shi <xushi@ucsd.edu>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

tude of recent price changes to identify overbought or oversold conditions.

- **Average True Range (ATR)** Reflects market volatility by computing a smoothed average of true ranges.

- **Log Return** The percentage change of tomorrow compared with today's. This is used as our training target.

The macroeconomic and commodities data are concatenated with the stock dataset based on the date for alignment, and the entire dataset is normalized using a Min-Max scaler.

2.2. Baseline LSTM

The baseline LSTM used in this paper is a classic version adapted from PyTorch. The LSTM is stacked by two layers with no pre-coded h_t and c_t weights.

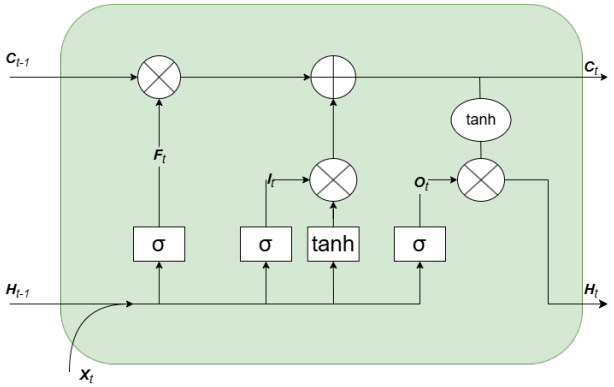

Figure 1. The classic LSTM neuron structure with gate mechanism, with F_t for forget gate, I_t for input gate, O_t for output gate.

2.3. Variant LSTM

The variant LSTM is derived from AMV-LSTM (Sang & Li, 2024). The structure proposed in Sang and Li's work couples the input and forget gates, enforcing a shared decision process that allows the model to update memory only when new input is present, thereby improving noise resistance and reducing the input gate's burden. Their AMV-LSTM demonstrated outstanding performance compared with classic LSTM and the peephole LSTM.

The variant LSTM used in this paper made some minor modifications, which removes the sigmoid gate from the previous output being passed along with the forget gate. The aim of this approach is to keep the coupled forget gate into the input gate while not having too much information being passed from the previous output.

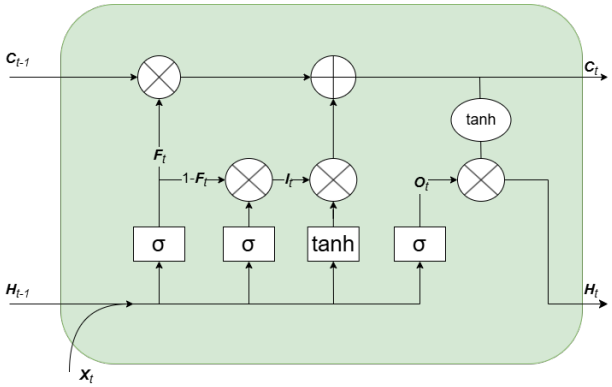

Figure 2. The variant LSTM neuron modified from AMV-LSTM.

2.4. Attention LSTM

The Attention LSTM proposed in this paper introduced a self-attention layer added after the Baseline LSTM mentioned in 2.2. The self-attention layer is constructed using a linear layer followed by a softmax layer. The purpose of the attention layer is to attend to all time steps, enabling improved recognition of long-term dependencies.

Let the output of the LSTM be a sequence of hidden vectors

$$\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_T] \in \mathbb{R}^{T \times d}.$$

A trainable linear layer projects each $\mathbf{h}_t$ into a scalar score:

$$e_t = \mathbf{w}^\top \mathbf{h}_t$$

The attention weights α_t are computed via softmax normalization:

$$\alpha_t = \frac{\exp(e_t)}{\sum_{k=1}^{T} \exp(e_k)}$$

The context vector $\mathbf{c}$ is then the weighted sum of all hidden states:

$$\mathbf{c} = \sum_{t=1}^{T} \alpha_t \mathbf{h}_t$$

The added attention layer is used to further capture event-related information on the macroeconomic data and modeling long-term dependencies which the LSTM did not catch.

2.5. Regression Construction and Optimizer Setting

All models mentioned earlier have a fully connected layer in the end with output dimension of 1 to convert them into a regression problem.

The optimizer chosen is Adam (Kingma & Ba, 2014), with weight decay being set to 10^{-5} for regularization purposes.

3. Experiment

3.1. Setup

The main dataset used in this paper is the daily stock information (Open, High, Low, Close, Volume) of Nvidia Corporation consisting of 3,623 daily observations, obtained from Yahoo Finance. The date ranges from May 09, 2015 to March 31, 2025. The stock's meta data including dividend and split are also used as features. The macroeconomic data was obtained from the Federal Reserve Bank of St. Louis. The daily commodities' price and the SP500 are also from Yahoo Finance. The commodities chosen are Lithium, Copper, Gold, and rare earth, all important raw materials for Nvidia's supply chain.

The indicators are calculated with window size of 5. The EMA indicators are calculated at 5, 10, 20, and 60-day intervals. Then the stock dataset is concatenated with the macroeconomic and commodities data, bringing 21 initial features for each day. The dataset is sliced with 9:1 train-test ratio and the Min-Max scaler is computed using the training dataset to prevent information leakage.

For all LSTM models, a sequence length of 7 is used and the training epoch number is set to 800 and the learning rate is 10^{-3}. Throughout multiple validation experiments, the model suffers serious overfitting problems after 1000 epochs.

Mean Square Error (MSE) is used as the evaluation criteria of model training. As a widely used loss function in regression tasks, MSE is also suitable for stock price prediction as it is robust against outliers and complex patterns.

The training data is further split into train and validation dataset using the Time Series split function from SKlearn. This function, unlike the usual splitting function, will not randomly pick one subset for validation. Instead, it will use the last portion of the sliced dataset, ensuring that no future information is leaked. During training, the algorithm will first undergo a 3-fold split to ensure its performance and generalization ability. Then it will be trained with the full training dataset. At last, the model will be set into evaluation mode and the test dataset will be fed to measure final performance.

Although the model is trained as a regression task, the final evaluation metric is the F1 score based on the confusion matrix of the direction of prediction. In the field of high-frequency trading, the future trend is more important than the precise future stock price. After the model predicts the log return, all predictions will be generalized into positive/negative boolean values to indicate whether the stock price will increase tomorrow.

Table 1. Confusion matrix and F1 score for the three LSTM models on Training Dataset.

LSTM	ACCURACY	PRECISION	RECALL	F1
BASELINE	0.8026	0.8329	0.7872	0.8094
VARIANT	0.8796	0.8597	0.9245	0.8911
ATTENTION	0.7666	0.8029	0.7444	0.7725

Table 2. Confusion matrix and F1 score for the three LSTM models on Testing Dataset.

LSTM	ACCURACY	PRECISION	RECALL	F1
BASELINE	0.5086	0.5494	0.6564	0.5981
VARIANT	0.4514	0.5882	0.0513	0.0943
ATTENTION	0.5314	0.5628	0.7128	0.6290

3.2. Results

On the training dataset, all three LSTM models perform adequately well, with the variant LSTM having the highest F1 score of 0.8911. However, good results on the training dataset do not guarantee a similar outcome in the testing dataset because of the overfitting problem. The Nvidia stock surged in the last two years, meaning the log return in the testing dataset is more dramatic compared with the training dataset, which contests the model's ability for generalization and robustness from overfitting.

On the testing dataset, the previously leading variant LSTM has an incredibly low F1 score of 0.09 due to its overfitting issue with the training dataset. The Attention has the highest F1 score of 0.629, beating the baseline LSTM by 2%. It is certain that the attention layer after the LSTM has successfully learned temporal long-term information which improves its performance in the test dataset.

Feature engineering and dataset construction are also proven to be vital for stock price prediction tasks. For datasets using only OHLCV data and indicators derived from it, the LSTM cannot pickup useful information. The model ended up randomly guessing with accuracy and F1 score fixed around 0.5.

4. Conclusion

Stock price prediction has always been a challenging task because of its almost random nonlinearity. Algorithms for this task aim for generalization and robustness for the future testing dataset. This paper proposes a combined approach using LSTM and an attention mechanism, as algorithms have the potential to memorize long-term trends and characteristics. This approach is proven to perform better on

the rapidly growing stock Nvidia, with higher F1 and accuracy scores compared with the baseline LSTM and a variant LSTM.

This paper also demonstrated the importance of data in time-series prediction. With the concatenated macroeconomic and commodities information, the model has higher probability of discovering underlying relationships between different assets. The increase of data dimension also helps the model overcome the overfitting problem, which is crucial for time series prediction.

In the future, approaches with augmented LSTM layers with attention mechanisms could potentially perform better compared with the current combination algorithm in the paper. Adding more data dimensions like semantic information or market events could also strengthen the model's ability.

References

Gers, F. A., Schmidhuber, J., and Cummins, F. Learning to forget: Continual prediction with lstm. *Neural computation*, 12(10):2451–2471, 2000.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Sang, S. and Li, L. A novel variant of lstm stock prediction method incorporating attention mechanism. *Mathematics*, 12(7), 2024. ISSN 2227-7390. doi: 10.3390/math12070945. URL https://www.mdpi.com/2227-7390/12/7/945.

Schuster, M. and Paliwal, K. K. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.